

JPO4/9517

PCT/JP2004/009517

22. 7. 2004

PA 1185207

RECEIVED

12 AUG 2004

WIPO

PCT

# THE UNITED STATES OF AMERICA

**TO ALL TO WHOM THESE PRESENTS SHALL COME:**

**UNITED STATES DEPARTMENT OF COMMERCE**

**United States Patent and Trademark Office**

**June 24, 2004**

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.**

**APPLICATION NUMBER: 60/483,228**

**FILING DATE: June 30, 2003**

**PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)**

**By Authority of the  
COMMISSIONER OF PATENTS AND TRADEMARKS**



*N. Williams*  
**N. WILLIAMS**  
**Certifying Officer**

PATENT APPLICATION SERIAL NO. \_\_\_\_\_

U.S. DEPARTMENT OF COMMERCE  
PATENT AND TRADEMARK OFFICE  
FEE RECORD SHEET

07/01/2003 H0UTERR1 00000069 60483228

01 FC:1005

160.00 DP

PTO-1556  
(5/87)

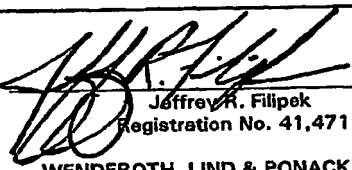

\*U S Government Printing Office 2002 — 489-287/69033

06/30/03

Please type a plus sign (+) inside this box - [ ]

PTO/SB/18 (2-8)  
Approved for use through 01-31-2001. OMB 0851-003  
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number**PROVISIONAL APPLICATION FOR PATENT COVER SHEET**

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.63(c)

INVENTOR(S)		
Given Name (first and middle (if any))	Family Name or Surname	Residence (City and either State or Foreign Country)
Hiroshi	YAHATA	25-15-202, Motomachi, Kadoma-shi, Osaka-fu 571-0057 Japan
<input checked="" type="checkbox"/> Additional inventors are being named on the <u>1</u> separately numbered sheets attached hereto		
TITLE OF THE INVENTION (280 characters max)		
INFORMATION RECORDING MEDIUM WITH STILL-PICTURE DATA RECORDED THEREIN		
CORRESPONDENCE ADDRESS		
Direct all correspondence to:		
<input type="checkbox"/> Customer Number _____		
Type Customer Number here		Place Customer Number Bar Code Label here
OR		
CORRESPONDENCE ADDRESS		By:  Jeffrey R. Filipek Registration No. 41,471 WENDEROTH, LIND & PONACK, L.L.P. 2033 "K" Street, N.W., Suite 800 Washington, D.C. 20006-1021 Phone: (202) 721-8200 Fax: (202) 721-8250 June 30, 2003
 000513 PATENT TRADEMARK OFFICE		
ENCLOSED APPLICATION PARTS (check all that apply)		
<input checked="" type="checkbox"/> Specification Number of Pages [61]		<input type="checkbox"/> CD(s), Number _____
<input checked="" type="checkbox"/> Drawing(s) Number of Sheets [48]		<input checked="" type="checkbox"/> Other (specify) <u>Patent Office Fee Transmittal Form</u>
METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)		
<input type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27.		
<input checked="" type="checkbox"/> A check or money order is enclosed to cover the filing fees		
<input type="checkbox"/> The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account No. <u>23-0975</u> .		Filing Fee Amount <u>\$160.00</u>
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.		
<input checked="" type="checkbox"/> No.		
<input type="checkbox"/> Yes, the name of the U.S. Government agency and the Government contract number are:		

Docket No.: PRO3 081Number 1 of 2THE COMMISSIONER IS AUTHORIZED  
TO ACCEPT ANY DEFICIENCY IN THE  
FILING OF THIS PAPER TO DEPOSIT  
ACCOUNT NO. 23-0975

# PROVISIONAL APPLICATION COVER SHEET

## Additional Page

PTO/SB/18 (2-95)  
Approved for use through 01-31-2001. OMB 0651-003  
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Docket No. PR03 0890

INVENTOR(S)/APPLICANT(S)		
Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)
Joseph	MCCROSSAN	472 San Jose Avenue #C, Burbank, CA 91501 U.S.A.
Tomoyuki	OKADA	1-8-19-303, Tomiomoto-machi, Nara-shi, Nara- ken 631-0078 Japan
Wataru	IKEDA	2-1-3-1205, Miyakojima-minamidori, Miyakojima-ku, Osaka-shi, Osaka-fu 534-0023 Japan

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of

Hiroshi YAHATA et al.

Serial No. NEW

Filed June 30, 2003

Attn: PROVISIONAL BRANCH

Attorney Docket No. PR03\_0890A

INFORMATION RECORDING MEDIUM WITH :  
STILL-PICTURE DATA RECORDED THEREIN

**PATENT OFFICE FEE TRANSMITTAL FORM**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

THIS PAPER IS TO BE DEPOSITED IN THE  
ACCOUNT NO. 23-0975

Sir:

Attached hereto is a check in the amount of \$160.00 to cover Patent Office fees relating to filing the following attached papers:

New provisional application ..... \$160.00

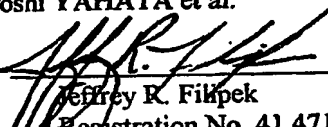
A duplicate copy of this paper is being submitted for use in the Accounting Division, Office of Finance.

*The Commissioner is authorized to charge any deficiency or to credit any overpayment associated with this communication to Deposit Account No. 23-0975, with the EXCEPTION of deficiencies in fees for multiple dependent claims in new applications.*

Respectfully submitted,

Hiroshi YAHATA et al.

By

  
Jeffrey R. Filipek  
Registration No. 41,471  
Attorney for Applicants

JRF/fs  
WENDEROTH, LIND & PONACK, L.L.P.  
2033 K St., N.W., Suite 800  
Washington, D.C. 20006-1021  
Telephone (202) 721-8200  
June 30, 2003

[Check No. 56398]  
PR03\_0890A

Title:

INFORMATION RECORDING MEDIUM WITH STILL-PICTURE DATA RECORDED THEREIN

【TITLE OF THE INVENTION】

静止画データを記録した情報記録媒体

【BACKGROUND OF THE INVENTION】

【Field of the Invention】

本発明は、動画データに付随して静止画データが記録された情報記録媒体に関する。

【Description of the Background Art】

従来の技術である、DVD-Videoディスク（以下単にDVDと呼ぶ）について説明する（例えば、特許文献1参照）。

図1は、DVDの構造を示した図である。図1の下段に示すように、DVDディスク上にはリードインからリードアウトまでの間に論理アドレス空間が設けられ、論理アドレス空間の先頭からファイルシステムのボリューム情報が記録され、続いて映像音声などのアプリケーションデータが記録されている。

ファイルシステムとは、ISO9660やUDF (Universal Disc Format) のことであり、ディスク上のデータをディレクトリまたはファイルと呼ばれる単位で表現する仕組みである。日常使っているPC（パーソナルコンピュータ）の場合でも、FATまたはNTFSと呼ばれるファイルシステムを通すことにより、ディレクトリやファイルという構造でハードディスクに記録されたデータがコンピュータ上で表現され、ユーザビリティを高めている。

DVDの場合、UDFおよびISO9660両方を使用しており（両方を合わせて「UDFブリッジ」と呼ぶ事がある）、UDFまたはISO9660どちらのファイルシステムドライバによってもデータの読み出しができるようになっている。勿論、書き換え型のDVDディスクであるDVD-RAM/R/RWでは、これらファイルシステムを介し、物理的にデータの読み、書き、削除が可能である。

DVD上に記録されたデータは、UDFブリッジを通して、図1左上に示すようなディレクトリまたはファイルとして見ることができる。ルートディレクトリ（図中

「ROOT」) の直下に「VIDEO\_TS」と呼ばれるディレクトリが置かれ、ここにDVDのアプリケーションデータが記録されている。アプリケーションデータは、複数のファイルとして記録され、主なファイルとして以下のものがある。

VIDEO_TS.IFO	ディスク再生制御情報ファイル
VTS_01_0.IFO	ビデオタイトルセット#1再生制御情報ファイル
VTS_01_0.VOB	ビデオタイトルセット#1ストリームファイル

.....

拡張子として2つの種類が規定されており、「IFO」は再生制御情報が記録されたファイルであって、「VOB」はAVデータであるMPEGストリームが記録されたファイルである。再生制御情報とは、DVDで採用されたインタラクティビティ（ユーザの操作に応じて再生を動的に変化させる技術）を実現するための情報や、メタデータのようなタイトルやAVストリームに付属する情報などのことである。また、DVDでは一般的に再生制御情報のことをナビゲーション情報と呼ぶことがある。

再生制御情報ファイルは、ディスク全体を管理する「VIDEO\_TS.IFO」と、個々のビデオタイトルセット（DVDでは複数のタイトル、言い換えれば異なる映画や異なるバージョンの映画を1枚のディスクに記録することが可能である。）毎の再生制御情報である「VTS\_01\_0.IFO」がある。ここで、ファイル名ボディにある「01」はビデオタイトルセットの番号を示しており、例えば、ビデオタイトルセット#2の場合は、「VTS\_02\_0.IFO」となる。

図1の右上部は、DVDのアプリケーション層でのDVDナビゲーション空間であり、前述した再生制御情報が展開された論理構造空間である。「VIDEO\_TS.IFO」内の情報は、VMGI (Video Manager Information) として、「VTS\_01\_0.IFO」または、他のビデオタイトルセット毎に存在する再生制御情報はVTSI (Video Title Set Information) としてDVDナビゲーション空間に展開される。

VTSIの中にはPGC (Program Chain) と呼ばれる再生シーケンスの情報であるPGCI (Program Chain Information) が記述されている。PGCIは、Cellの集合とコマンドと呼ばれる一種のプログラミング情報によって構成されている。Cell自身はVOB (Video Objectの略であり、MPEGストリームを指す) の一部区間または全部区間の集合であり、Cellの再生は、当該VOBのCellによって指定された区間を再生することを意味している。

コマンドは、DVDの仮想マシンによって処理されるものであり、ブラウザ上で実行されるJavaスクリプトなどに近いものである。しかしながらJavaスクリプトが論理演算の他にウィンドウやブラウザの制御 (例えば、新しいブラウザのウィンドウを開くなど) を行うのに対して、DVDのコマンドは、論理演算の他にAVタイトルの再生制御、例えば、再生するチャプタの指定などを実行するだけのものである点で異なっている。

Cellはディスク上に記録されているVOBの開始および終了アドレス (ディスク上での論理記録アドレス) をその内部情報として有しており、プレーヤは、Cellに記述されたVOBの開始および終了アドレス情報を使ってデータの読み出し、再生を実行する。

図2はAVストリーム中に埋め込まれているナビゲーション情報を説明する概略図である。DVDの特長であるインタラクティビティは前述した「VIDEO\_TS.IFO」や「VTS\_01\_0.IFO」などに記録されているナビゲーション情報だけによって実現されているのではなく、幾つかの重要な情報はナビゲーションパック (ナビパックまたは、NV\_PCKと称する) と呼ばれる専用キャリアを使いVOB内に映像、音声データと一緒に多重化されている。

ここでは簡単なインタラクティビティの例としてメニューを説明する。メニュー画面上には、幾つかのボタンが現れ、夫々のボタンには当該ボタンが選択実行



された時の処理が定義されている。また、メニュー上では一つのボタンが選択されており（ハイライトによって選択ボタン上に半透明色がオーバーレイされており該ボタンが選択状態であることをユーザーに示す）、ユーザは、リモコンの上下左右キーを使って、選択状態のボタンを上下左右の何れかのボタンに移動させることが出来る。リモコンの上下左右キーを使って、選択実行したいボタンまでハイライトを移動させ、決定する（決定キーを押す）ことによって対応するコマンドのプログラムが実行される。一般的には対応するタイトルやチャプタの再生がコマンドによって実行されている。

図2の左上部はNV\_PCK内に格納される制御情報の概要を示している。

NV\_PCK内には、ハイライトカラー情報と個々のボタン情報などが含まれている。ハイライトカラー情報には、カラーパレット情報が記述され、オーバーレイ表示されるハイライトの半透明色が指定される。ボタン情報には、個々のボタンの位置情報である矩形領域情報と、当該ボタンから他のボタンへの移動情報（ユーザの上下左右キー操作夫々に対応する移動先ボタンの指定）と、ボタンコマンド情報（当該ボタンが決定された時に実行されるコマンド）が記述されている。

メニュー上のハイライトは、図2の中央右上部に示すように、オーバーレイ画像として作られる。オーバーレイ画像は、ボタン情報の矩形領域情報にカラーパレット情報の色をつけた物である。このオーバーレイ画像は図2の右部に示す背景画像と合成されて画面上に表示される。

上述のようにして、DVDではメニューを実現している。また、何故、ナビゲーションデータの一部をNV\_PCKを使ってストリーム中に埋め込んでいるのは、ストリームと同期して動的にメニュー情報を更新したり（例えば、映画再生の途中5分～10分の間にだけメニューが表示されるなど）、同期タイミングが問題となりやすいアプリケーションの場合でも、問題なく実現できるようにしたためである。また、もう一つの大きな理由は、NV\_PCKには特殊再生を支援するための情報を格納し、DVD再生時の早送り、巻き戻しなどの非通常再生時にも円滑にAVデー

タをデコードし再生させる等、ユーザーの操作性を向上させるためである。

図3は、DVDのストリームであるVOBのイメージである。図に示すように、映像、音声、字幕などのデータ（A段）は、MPEGシステム規格（ISO/IEC13818-1）に基づいて、パケットおよびパック化し（B段）、夫々を多重化して1本のMPEGプログラムストリームにしている（C段）。また、前述した通りインタラクティブを実現するためのボタンコマンドを含んだNV\_PCKも一緒に多重化をされている。

MPEGシステムの多重化の特徴は、多重化する個々のデータは、そのデコード順に基づくビット列になっているが、多重化されるデータ間、即ち、映像、音声、字幕の間は必ずしも再生順、言い換えればデコード順に基づいてビット列が形成されている訳ではない。これは多重化したMPEGシステムストリームのデコーダモデル（一般にSystem Target Decoder、またはSTDと呼ばれる（図3のD段））が多重化を解いた後に個々のエレメンタリーストリームに対応するデコーダバッファを持ち、デコードタイミングまでに一時的にデータを蓄積している事に由来している。例えばDVD-Videoで規定されるデコーダバッファは、個々のエレメンタリーストリーム毎にサイズが異なり、映像に対しては、232KB、音声に対しては4KB、字幕に対しては52KBを夫々有している。

即ち、映像データと並んで多重化されている字幕データが必ずしも同一タイミングでデコードもしくは再生されているわけでは無い。

#### 【特許文献1】

特許第3128220号

しかしながら従来のDVDのメニュー画面におけるボタンは、背景にボタン画像を埋め込み、ハイライト部分の色だけを変えて実現していた。そのため、ボタンイメージとフォーカスが当たったときおよびボタンを選択したときに表示するハイライトイメージを別々に作成しなくてはならず、オーサリング（DVDのタイトル制作）が困難であった。

ハイライトに許される色数も限定されており、背景に組み込まれたボタン画像

が綺麗であっても、ハイライトは色数の少ないあまり綺麗ではないイメージデータであった。

また、イメージデータに関しても、ハイライトの制御情報と実際のイメージデータは別々に格納されているため、オーサリングが困難であった。

従来のDVDのハイライトでは、数字キーによるボタン選択は連番でしか設定できないといった欠点があった。そのため、プロ野球選手の背番号ごとのメニューなどといったアプリケーションを作成できなかった。

従来のDVDでは、カーソルキーを移動した際、あるいはボタンにフォーカスが当たったときおよび選択したときに、これを示す効果音がなかった。

イメージデータのデコードには時間がかかり、迅速な表示は期待できなかった。

#### 【SUMMARY OF THE INVENTION】

本発明の目的は、従来のDVDでは非常に困難だったことに加え、実現不可能な構成が存在することを解決し、新しいアプリケーションおよびハイライト構造を実現し、ユーザーに新たなサービスの実現が可能な情報記録媒体を提供することである。

上記課題を解決するため、本発明に係る情報記録媒体は、少なくとも一つ以上の静止画データが記録しており、

静止画データは、一つ若しくは複数の静止画データを格納する論理ブロックに格納され、前記論理ブロック先頭には論理ブロックの制御情報が配され、前記制御情報には、当該論理ブロックのタイプを示す識別情報と、当該論理ブロック内に格納された静止画データの表示管理情報とが配置され、前記識別情報が第一の論理ブロックタイプを示している場合、表示管理情報は、該論理ブロック内に格納される全ての静止画データの表示位置を示し、前記識別情報が第二の論理ブロックタイプを示している場合、表示管理情報は、該論理ブロック内に格納される全て、若しくは、一部の静止画データの表示位置を示すものである。

本発明では、メニュー画面などにおけるボタンイメージとその制御情報をカプセル化しているので、よりハイライトを扱いやすくする効果がある。

ここで前記識別情報が第一の論理ブロックタイプを示している場合、識別情報

を含む制御情報には、該論理ブロックに格納される全ての静止画データに対する識別番号を割り振っておいてもよい(2)。

また、ボタンイメージの格納方法を最適化することで、デコーダの処理遅延を隠蔽化することができる。

その格納方法の最適化には、以下の(3)～(16)のものがある。

(3)前記識別情報が第一の論理ブロックタイプを示している場合、

当該論理ブロックに、第一の静止画データグループを先頭から配置する。第一の静止画データグループは、ユーザーからの操作が無くとも表示される静止画データからなる。

(4)前記第一の静止画データグループにおいて、静止画データを、前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って配列させる。

(5)第1の静止画データグループにおいて、表示位置が同一であって、スキャンライン順が同順となる静止画データを、前記識別番号の小さい順に先頭から配置させる。

(6)前記第一の論理グループにおいて、第一の静止画データグループには、

前記第一の論理ブロックに格納される静止画データであって、次に一回のユーザーからの操作に応じて表示される可能性のある静止画データからなる第二の静止画データグループを、後続させる。

(7)前記第2の静止画データグループにおいて、静止画データを、前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って配列させる。

(8)第2の静止画データグループにおいて、表示位置が同一であって、スキャンライン順が同順となる静止画データを、前記識別番号の小さい順に先頭から配置させる。

(9)前記識別情報が第一の論理ブロックタイプを示している場合、その第一の論理ブロックに格納される静止画データであって、前記第一の静止画データグループおよび前記第二の静止画データグループに属さない第三の静止画データグループの静止画データを、前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って先頭から配置させる。

(10)第三の静止画データグループにおいて、表示位置が同一であり、スキャンライン順が同順となる静止画データを、前記識別番号の小さい順に先頭から配置させる。

(11)第二の静止画データグループに属する静止画データを、前記制御情報に記述された前記識別番号の小さい順に配置させる。

(12)前記第三の静止画データグループ内の静止画データを、前記制御情報に記述された前記識別番号の小さい順に先頭から配置させる。

(13)前記識別情報が第二の論理ブロックタイプを示している場合、その論理ブロックの制御情報には、該論理ブロックに格納される全てもしくは一部の静止画データに対する識別番号が割り振る。

(14)前記識別情報が第二の論理ブロックタイプを示している場合、その論理ブロックには、前記識別番号が割り振られた全ての静止画データから構成される第四の静止画データグループを、先頭から配置させる。

(15)前記第四の静止画データグループに格納される静止画データは、前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って先頭から配置させる。

(16)前記識別情報が第二の論理ブロックタイプを示している場合、その論理ブ

ロックには、前記識別番号が割り振られていない全ての静止画データから構成される第五の静止画データグループが、前記第四の静止画データグループに続いて配置させる。

更に、静止画と動画の同期再生モデルを規定することで、機器間互換性を確保することができる。

#### 【BRIEF DESCRIPTION OF THE DRAWINGS】

These and other objects, advantages and features of the invention will become apparent from the following description thereof taken in conjunction with the accompanying drawings which illustrate a specific embodiment of the invention. In the Drawings:

- 【図1】 DVDの構成図である。
- 【図2】 ハイライトの構成図である。
- 【図3】 DVDでの多重化の例を示す図である。
- 【図4】 HD-DVDのデータ階層図である。
- 【図5】 HD-DVD上の論理空間の構成図である。
- 【図6】 HD-DVDプレーヤの概要ブロック図である。
- 【図7】 HD-DVDプレーヤの構成ブロック図である。
- 【図8】 HD-DVDのアプリケーション空間の説明図である。
- 【図9】 MPEGストリーム (VOB) の構成図である。
- 【図10】 パックの構成図である。
- 【図11】 AVストリームとプレーヤ構成の関係を説明する図である。
- 【図12】 トラックバッファへのAVデータ連続供給モデル図である。
- 【図13】 VOB情報ファイル構成図である。
- 【図14】 タイムマップの説明図である。
- 【図15】 タイムマップを使ったアドレス情報取得方法説明図である。
- 【図16】 プレイリストファイルの構成図である。
- 【図17】 プレイリストに対応するプログラムファイルの構成図である。

- 【図18】BDディスク全体管理情報ファイルの構成図である。
- 【図19】グローバルイベントハンドラを記録するファイルの構成図である。
- 【図20】タイムイベントの例を説明する図である。
- 【図21】ユーザイベントの例を説明する図である。
- 【図22】グローバルイベントハンドラの例を説明する図である。
- 【図23】仮想マシンの構成図である。
- 【図24】プレーヤ変数テーブルの図である。
- 【図25】イベントハンドラ（タイムイベント）の例を示す図である。
- 【図26】イベントハンドラ（ユーザイベント）の例を示す図である。
- 【図27】プレーヤの基本処理のフローチャートである。
- 【図28】プレイリスト再生処理のフローチャートである。
- 【図29】イベント処理のフローチャートである。
- 【図30】字幕処理のフローチャートである。
- 【図31】BDにおけるハイライトを説明する図である。
- 【図32】カプセル化されたハイライトユニットの構造図である。
- 【図33】ハイライトユニットのボタン制御情報の構造図である。
- 【図34】ページ属性情報の各属性情報を説明する図である。
- 【図35】ボタン属性情報の各属性情報を説明する図である。
- 【図36】ハイライトユニットの配置を説明する図である。
- 【図37】duplicateハイライトユニットの配置を説明する図である。
- 【図38】status\_of\_Highlightの利用方法を説明する図である。
- 【図39】効果音データもカプセル化されたハイライトユニットの構造図である。
- 【図40】ハイライトスチルを説明する図である。
- 【図41】ユーザー操作と表示ボタンの遷移関係を説明する図である。
- 【図42】ボタンイメージの転送順序の関係を説明する図である。
- 【図43】ボタンイメージを格納するPESパケットの構造を説明する図である。
- 【図44】ボタンイメージを格納するPESパケットの構造を説明する図である。
- 【図45】動画と静止画を同期させるデコーダモデルを説明する図である。

【図４６】字幕用ハイライトを説明する図である。

【図４７】字幕用ハイライトの静止画データ格納順序を説明する図である。

【図４８】メニュー用ハイライトの静止画データ格納順序を説明する図である。

## 【DESCRIPTION OF THE PREFERRED EMBODIMENT(S)】

### （実施例１）

まず最初に本発明の第１の実施の形態について説明する。

#### （ディスク上の論理データ構造）

図４は、次世代DVD（以降、BDと称する）の構成、特にディスク媒体であるBDディスク（１０４）と、ディスクに記録されているデータ（１０１、１０２、１０３）の構成を示す図である。BDディスク（１０４）に記録されるデータは、AVデータ（１０３）と、AVデータに関する管理情報およびAV再生シーケンスなどのBD管理情報（１０２）と、インタラクティブを実現するBD再生プログラム（１０１）である。本実施の形態では、説明の都合上、映画のAVコンテンツを再生するためのAVアプリケーションを主眼においてのBDディスクの説明を行うが、他の用途として用いても勿論同様である。

図５は、上述したBDディスクに記録されている論理データのディレクトリ・ファイル構成を示した図である。BDディスクは、他の光ディスク、例えばDVDやCDなどと同様にその内周から外周に向けてらせん状に記録領域を持ち、内周のリード・インと外周のリード・アウトの間に論理データを記録できる論理アドレス空間を有している。また、リード・インの内側にはBCA（Burst Cutting Area）と呼ばれるドライブでしか読み出せない特別な領域がある。この領域はアプリケーションから読み出せないため、例えば著作権保護技術などに利用されることがある。

論理アドレス空間には、ファイルシステム情報（ボリューム）を先頭に映像データなどのアプリケーションデータが記録されている。ファイルシステムとは従来技術で説明した通り、UDFやISO9660などのことであり、通常のPCと同じように



記録されている論理データをディレクトリ、ファイル構造を使って読み出しする事が可能になっている。

本実施例の場合、BDディスク上のディレクトリ、ファイル構造は、ルートディレクトリ（ROOT）直下にBDVIDEOディレクトリが置かれている。このディレクトリはBDで扱うAVコンテンツや管理情報などのデータ（図4で説明した101、102、103）が格納されているディレクトリである。

BDVIDEOディレクトリの下には、次の7種類のファイルが記録されている。

**BD.INFO（ファイル名固定）**

「BD管理情報」の一つであり、BDディスク全体に関する情報を記録したファイルである。BDプレーヤは最初にこのファイルを読み出す。

**BD.PROG（ファイル名固定）**

「BD再生プログラム」の一つであり、BDディスク全体に関わる再生制御情報を記録したファイルである。

**XXX.PL（「XXX」は可変、拡張子「PL」は固定）**

「BD管理情報」の一つであり、シナリオ（再生シーケンス）であるプレイリスト情報を記録したファイルである。プレイリスト毎に1つのファイルを持っている。

**XXX.PROG（「XXX」は可変、拡張子「PL」は固定）**

「BD再生プログラム」の一つであり、前述したプレイリスト毎の再生制御情報を記録したファイルである。プレイリストとの対応はファイルボディ名（「XXX」が一致する）によって識別される。

**YYY.VOB（「YYY」は可変、拡張子「VOB」は固定）**

「AVデータ」の一つであり、VOB（従来例で説明したVOBと同じ）を記録したファイルである。VOB毎に1つのファイルを持っている。

YYY.VOBI（「YYY」は可変、拡張子「VOBI」は固定）

「BD管理情報」の一つであり、AVデータであるVOBに関わるストリーム管理情報を記録したファイルである。VOBとの対応はファイルボディ名（「YYY」が一致する）によって識別される。

ZZZ.PNG（「ZZZ」は可変、拡張子「PNG」は固定）

「AVデータ」の一つであり、字幕およびメニューを構成するためのイメージデータPNG（W3Cによって標準化された画像フォーマットであり「ピング」と読む）を記録したファイルである。1つのPNGイメージ毎に1つのファイルを持つ。

（プレーヤの構成）

次に、前述したBDディスクを再生するプレーヤの構成について図6および図7を用いて説明する。

図6は、プレーヤの大まかな機能構成を示すブロック図である。

BDディスク（201）上のデータは、光ピックアップ（202）を通して読み出される。読み出されたデータは夫々のデータの種別に応じて専用のメモリに転送される。BD再生プログラム（「BD.PROG」または「XXX.PROG」ファイルの中身）はプログラム記録メモリ（203）に、BD管理情報（「BD.INFO」、「XXX.PL」または「YYY.VOBI」）は管理情報記録メモリ（204）に、AVデータ（「YYY.VOB」または「ZZZ.PNG」）はAV記録メモリ（205）に夫々転送される。

プログラム記録メモリ（203）に記録されたBD再生プログラムはプログラム処理部（206）によって、管理情報記録メモリ（204）に記録されたBD管理情報は管理情報処理部（207）によって、また、AV記録メモリ（205）に記録されたAVデータはプレゼンテーション処理部（208）によって夫々処理され

る。

プログラム処理部（２０６）は、管理情報処理部（２０７）より再生するプレイリストの情報やプログラムの実行タイミングなどのイベント情報を受け取りプログラムの処理を行う。また、プログラムでは再生するプレイリストを動的に変える事が可能であり、この場合は管理情報処理部（２０７）に対してプレイリストの再生命令を送ることで実現する。プログラム処理部（２０６）は、ユーザからのイベント、即ちリモコンキーからのリクエストを受け、ユーザイベントに対応するプログラムがある場合は、それを実行する。

管理情報処理部（２０７）は、プログラム処理部（２０６）の指示を受け、対応するプレイリストおよびプレイリストに対応したVOBの管理情報を解析し、プレゼンテーション処理部（２０８）に対象となるAVデータの再生を指示する。また、管理情報処理部（２０７）は、プレゼンテーション処理部（２０８）より基準時刻情報を受け取り、時刻情報に基づいてプレゼンテーション処理部（２０８）にAVデータ再生の停止指示を行い、また、プログラム処理部（２０６）に対してプログラム実行タイミングを示すイベントを生成する。

プレゼンテーション処理部（２０８）は、映像、音声、字幕／イメージ（静止画）の夫々に対応するデコーダを持ち、管理情報処理部（２０７）からの指示に従い、AVデータのデコードおよび出力を行う。映像データ、字幕／イメージの場合は、デコード後に夫々の専用プレーン、ビデオプレーン（２１０）およびイメージプレーン（２０９）に描画され、合成処理部（２１１）によって映像の合成処理が行われTVなどの表示デバイスへ出力される。

このように図６に示すように、BDプレーヤは図４で示したBDディスクに記録されているデータ構成に基づいた機器構成をとっている。

図７は前述したプレーヤ構成を詳細化したブロック図である。図７では、AV記

録メモリ（２０５）はイメージメモリ（３０８）とトラックバッファ（３０９）に、プログラム処理部（２０６）はプログラムプロセッサ（３０２）とUOPマネージャ（３０３）に、管理情報処理部（２０７）はシナリオプロセッサ（３０５）とプレゼンテーションコントローラ（３０６）に、プレゼンテーション処理部（２０８）はクロック（３０７）、デマルチプレクサ（３１０）、イメージプロセッサ（３１１）、ビデオプロセッサ（３１２）とサウンドプロセッサ（３１３）に夫々対応／展開している。

BDディスク（２０１）から読み出されたVOBデータ（MPEGストリーム）はトラックバッファ（３０９）に、イメージデータ（PNG）はイメージメモリ（３０８）に夫々記録される。デマルチプレクサ（３１０）がクロック（３０７）の時刻に基づき、トラックバッファ（３０９）に記録されたVOBデータを抜き出し、映像データをビデオプロセッサ（３１２）に音声データをサウンドプロセッサ（３１３）に夫々送り込む。ビデオプロセッサ（３１２）およびサウンドプロセッサ（３１３）は夫々MPEGシステム規格で定める通りに、デコーダバッファとデコーダから夫々構成されている。即ち、デマルチプレクサ（３１０）から送りこまれる映像、音声夫々のデータは、夫々のデコーダバッファに一時的に記録され、クロック（３０７）に従い個々のデコーダでデコード処理される。

イメージメモリ（３０８）に記録されたPNGは、次の２つの処理方法がある。

イメージデータが字幕用の場合は、プレゼンテーションコントローラ（３０６）によってデコードタイミングが指示される。クロック（３０７）からの時刻情報をシナリオプロセッサ（３０５）が一旦受け、適切な字幕表示が行えるように、字幕表示時刻（開始および終了）になればプレゼンテーションコントローラ（３０６）に対して字幕の表示、非表示の指示を出す。プレゼンテーションコントローラ（３０６）からデコード／表示の指示を受けたイメージプロセッサ（３１１）は対応するPNGデータをイメージメモリ（３０８）から抜き出し、デコードし、イメージプレーン（３１４）に描画する。

次に、イメージデータがメニュー用の場合は、プログラムプロセッサ（３０２）によってデコードタイミングが指示される。プログラムプロセッサ（３０２）が何時イメージのデコードを指示するかは、プログラムプロセッサ（３０２）が処理しているBDプログラムに因るものであって一概には決まらない。

イメージデータおよび映像データは、図６で説明したように夫々デコード後にイメージプレーン（３１４）、ビデオプレーン（３１５）に出力され、合成処理部（３１６）によって合成後出力される。

BDディスク（２０１）から読み出された管理情報（シナリオ、AV管理情報）は、管理情報記録メモリ（３０４）に格納されるが、シナリオ情報（「BD.INFO」および「XXX.PL」）はシナリオプロセッサ（３０５）へ読み込み処理される。また、AV管理情報（「YYY.VOBI」）はプレゼンテーションコントローラ（３０６）によって読み出され処理される。

シナリオプロセッサ（３０５）は、プレイリストの情報を解析し、プレイリストによって参照されているVOBとその再生位置をプレゼンテーションコントローラ（３０６）に指示し、プレゼンテーションコントローラ（３０６）は対象となるVOBの管理情報（「YYY.VOBI」）を解析して、対象となるVOBを読み出すようにドライブコントローラ（３１７）に指示を出す。

ドライブコントローラ（３１７）はプレゼンテーションコントローラ（３０６）の指示に従い、光ピックアップを移動させ、対象となるAVデータの読み出しを行う。読み出されたAVデータは、前述したようにイメージメモリ（３０８）またはトラックバッファ（３０９）に読み出される。

また、シナリオプロセッサ（３０５）は、クロック（３０７）の時刻を監視し、管理情報で設定されているタイミングでイベントをプログラムプロセッサ（３０２）に投げる。

プログラム記録メモリ（301）に記録されたBDプログラム（「BD.PROG」または「XXX.PROG」）は、プログラムプロセッサ302によって実行処理される。プログラムプロセッサ（302）がBDプログラムを処理するのは、シナリオプロセッサ（305）からイベントが送られてきた場合か、UOPマネージャ（303）からイベントが送られたきた場合である。UOPマネージャ（303）は、ユーザからリモコンキーによってリクエストが送られてきた場合に、プログラムプロセッサ（302）に対するイベントを生成する。

#### （アプリケーション空間）

図8は、BDのアプリケーション空間を示す図である。

BDのアプリケーション空間では、プレイリスト（PlayList）が一つの再生単位になっている。プレイリストはセル（Cell）の連結で、連結の順序により決定される再生シーケンスである静的なシナリオと、プログラムによって記述される動的なシナリオを有している。プログラムによる動的なシナリオが無い限り、プレイリストは個々のセルを順に再生するだけであり、また、全てのセルの再生を終了した時点でプレイリストの再生は終了する。一方で、プログラムは、プレイリストを超えての再生記述や、ユーザ選択またはプレーヤの状態によって再生する対象を動的に変えることが可能である。典型的な例としてはメニューがあげられる。BDの場合、メニューとはユーザの選択によって再生するシナリオと定義でき、プログラムによってプレイリストを動的に選択することである。

ここで言うプログラムとは、時間イベントまたはユーザイベントによって実行されるイベントハンドラの事である。

時間イベントは、プレイリスト中に埋め込まれた時刻情報に基づいて生成されるイベントである。図7で説明したシナリオプロセッサ（305）からプログラムプロセッサ（302）に送られるイベントがこれに相当する。時間イベントが発行されると、プログラムプロセッサ（302）はIDによって対応付けられるイベントハンドラを実行処理する。前述した通り、実行されるプログラムが他のプ

レイリストの再生を指示することが可能であり、この場合には、現在再生されているプレイリストの再生は中止され、指定されたプレイリストの再生へと遷移する。

ユーザイベントは、ユーザのリモコンキー操作によって生成されるイベントである。ユーザイベントは大きく2つのタイプに分けられる。一つ目は、カーソルキー（「上」「下」「左」「右」キー）または「決定」キーの操作によって生成されるメニュー選択のイベントである。メニュー選択のイベントに対応するイベントハンドラはプレイリスト内の限られた期間でのみ有効であり（プレイリストの情報として、個々のイベントハンドラの有効期間が設定されている）、リモコンの「上」「下」「左」「右」キーまたは「決定」キーが押された時に有効なイベントハンドラを検索して、有効なイベントハンドラがある場合は当該イベントハンドラが実行処理される。他の場合は、メニュー選択のイベントは無視されることになる。

二つ目のユーザイベントは、「メニュー」キーの操作によって生成されるメニュー呼び出しのイベントである。メニュー呼び出しのイベントが生成されると、グローバルイベントハンドラが呼ばれる。グローバルイベントハンドラはプレイリストに依存せず、常に有効なイベントハンドラである。この機能を使うことにより、DVDのメニューコール（タイトル再生中に音声、字幕メニューなどと呼び出し、音声または字幕を変更後に中断した地点からのタイトル再生を実行する機能等）を実装することができる。

プレイリストで静的シナリオを構成する単位であるセル（Cell）はVOB（MPEGストリーム）の全部または一部の再生区間を参照したものである。セルはVOB内の再生区間を開始、終了時刻の情報として持っている。個々のVOBと一対になっているVOB管理情報（VOBI）は、その内部にデータの再生時刻に対応した記録アドレスのテーブル情報であるタイムマップ（Time MapまたはTM）を有しており、このタイムマップによって前述したVOBの再生、終了時刻をVOB内（即ち対象とな

るファイル「YYY.VOB」内)での読み出し開始アドレスおよび終了アドレスを導き出すことが可能である。なおタイムマップの詳細は後述する。

#### (VOBの詳細)

図9は、本実施例で使用するMPEGストリーム (VOB) の構成図である。

図9に示すように、VOBは複数のVOBU (Video Object Unit) によって構成されている。VOBUは、MPEGビデオストリームで言うGOP (Group Of Pictures) を基準として、音声データも含んだ多重化ストリームとしての一再生単位である。VOBUは0.4秒から1.0秒のビデオ再生時間を持ち、通常は0.5秒程度の再生時間を持っている。つまり、多くのケースで1GOPには15フレーム程度のフレームが格納されている。(NTSCの場合)

VOBUは、その内部にビデオパック (V\_PCK) とオーディオパック (A\_PCK) を有している。各パックは1セクタと同じサイズであり、本実施例の場合は2KB単位で構成されている。

図10は、パックの構成を示した図である。

図10に示すように、ビデオデータおよびオーディオデータといったエレメンタリデータは、ペイロード (PES Packet Payload) と呼ばれるパケット (PES Packet) のデータ格納領域に先頭から順次入れられていく。ペイロードにはパケットヘッダ (PES Packet Header) が付けられ1つのパケット (PES Packet) を構成する。パケットヘッダには、ペイロードに格納してあるデータがどのストリームなのかを識別するためのID (stream\_id) と、当該ペイロードのデコードおよび表示時刻情報であるタイムスタンプ、DTS (Decoding Time Stamp) およびPTS (Presentation Time Stamp) が記録される。PTS/DTSは必ずしも全てのパケットヘッダに記録されている訳ではなく、MPEGによってルールが規定されている。ルールの詳細についてはMPEGシステム (ISO/IEC13818-1) 規格書に記述されているので省略する。



パケットには更にヘッダ (Pack Header) が付けられ、パックを構成する。パックヘッダには、当該パックがいつデマルチプレクサを通過し、個々のエレメンタリストリームのデコーダバッファに入力されるかを示すタイムスタンプSCR (System Clock Reference) が記録されている。

#### (VOBのインターリーブ記録)

次に図11および図12を用いてVOBファイルのインターリーブ記録について説明する。

図11上段は、前述したプレーヤ構成図の一部である。図の通り、BDディスク上のデータは、光ピックアップを通してVOB即ちMPEGストリームであればトラックバッファへ入力され、PNG即ちイメージデータであればイメージメモリへと入力される。

トラックバッファはFIFOであり、入力されたVOBのデータは入力された順にデマルチプレクサへと送られる。この時、前述したSCRに従って個々のパックはトラックバッファから引き抜かれデマルチプレクサを介してビデオプロセッサまたはサウンドプロセッサへとデータが送り届けられる。一方で、イメージデータの場合は、どのイメージを描画するかはプレゼンテーションコントローラによって指示される。また、描画に使ったイメージデータは、字幕用イメージデータの場合は同時にイメージメモリから削除されるが、メニュー用のイメージデータの場合は、そのメニュー描画中はイメージメモリ内にそのまま残される。これはメニューの描画はユーザ操作に依存しており、ユーザーの操作に追従してメニューの一部分を再表示もしくは異なるイメージに置き換えることがあり、その際に再表示される部分のイメージデータをデコードし易くするためである。

図11下段は、BDディスク上でのVOBファイルおよびPNGファイルのインターリーブ記録を示す図である。一般的にROM、例えばCD-ROMやDVD-ROMの場合、一連の連続再生単位となるAVデータは連続記録されている。これは、連続記録されてい

る限り、ドライブは順次データを読み出し、デコードに送り届けるだけで良いが、連続データが分断されてディスク上に離散配置されている場合は、個々の連続区間の間でシーク操作が入ることになり、この間データの読み出しが止まることになり、データの供給が止まる可能性があるからである。BDの場合も同様に、VOBファイルは連続領域に記録することができる方が望ましいが、例えば字幕データのようにVOBに記録されている映像データと同期して再生されるデータがあり、VOBファイルと同様に字幕データも何らかの方法によってBDディスクから読み出す事が必要になる。

字幕データの読み出し方法の一手段として、VOBの再生開始前に一まとめで字幕用のイメージデータ（PNGファイル）を読み出してしまう方法がある。しかしながら、この場合には大量のメモリが必要となり、非現実的である。

そこで、本実施の形態では、VOBファイルを幾つかのブロックに分けて、イメージデータとインターリーブ記録する方式を使用している。図11下段はそのインターリーブ記録を説明した図である。

VOBファイルとイメージデータを適切にインターリーブ配置することで、前述したような大量の一時記録メモリ無しに、必要なタイミングでイメージデータをイメージメモリに格納することが可能になる。しかしながらイメージデータを読み出している際には、VOBデータの読み込みは当然のことながら停止することになる。

図12は、この問題を解決するトラックバッファを使ったVOBデータ連続供給モデルを説明する図である。

既に説明したように、VOBのデータは、一旦トラックバッファに蓄積される。トラックバッファへのデータ入力レート（ $V_a$ ）とトラックバッファからのデータ出力レート（ $V_b$ ）の間に差（ $V_a > V_b$ ）を設けると、BDディスクからデータを読み出し続けている限り、トラックバッファのデータ蓄積量は増加をしていくことに

なる。

図12の上段に記すようにVOBの一連続記録領域が論理アドレスの" a1" から" a2" まで続くとする。" a2" から" a3" の間は、イメージデータが記録されていて、VOBデータの読み出しが行えない区間であるとする。

図12の下段は、トラックバッファの内部を示す図である。横軸が時間、縦軸がトラックバッファ内部に蓄積されているデータ量を示している。時刻" t1" がVOBの一連続記録領域の開始点である" a1" の読み出しを開始した時刻を示している。この時刻以降、トラックバッファにはレート $V_a - V_b$ でデータが蓄積されていくことになる。このレートは言うまでもなくトラックバッファの入出力レートの差である。時刻" t2" は一連続記録領域の終了点である" a2" のデータを読み込む時刻である。即ち時刻" t1" から" t2" の間レート $V_a - V_b$ でトラックバッファ内はデータ量が増加していき、時刻" t2" でのデータ蓄積量は $B(t2)$ は下式によって求めることができる。

$$B(t2) = (V_a - V_b) \times (t2 - t1) \quad (\text{式1})$$

この後、BDディスク上のアドレス" a3" まではイメージデータが続くため、トラックバッファへの入力は0となり、出力レートである"  $V_b$ " でトラックバッファ内のデータ量は減少していくことになる。これは読み出し位置" a3" まで、時刻でいう" t3" までになる。

ここで大事なことは、時刻" t3" より前にトラックバッファに蓄積されているデータ量が0になると、デコーダへ供給するVOBのデータが無くなってしまい、VOBの再生がストップしてしまう可能性がある。しかしながら、時刻" t3" でトラックバッファにデータが残っている場合には、VOBの再生がストップすることなく連続できることを意味している。

この条件は下式によって示すことができる。

$$B(t_2) \geq -V_b \times (t_3 - t_2) \quad (\text{式 2})$$

即ち、式2を満たすようにイメージデータ（非VOBデータ）の配置を決めればよい事になる。

（ナビゲーションデータ構造）

図13から図19を用いて、BDのナビゲーションデータ（BD管理情報）構造について説明をする。

図13は、VOB管理情報情報ファイル（“YYY.VOBI”）の内部構造を示した図である。

VOB管理情報は、当該VOBのストリーム属性情報（Attribute）とタイムマップ（TMAP）を有している。ストリーム属性は、ビデオ属性（Video）、オーディオ属性（Audio#0～Audio#m）個々に持つ構成となっている。特にオーディオストリームの場合は、VOBが複数本のオーディオストリームを同時に持つことができることから、オーディオストリーム数（Number）によって、データフィールドの有無を示している。

下記はビデオ属性（Video）の持つフィールドと夫々が持ち得る値である。

圧縮方式（Coding）：

MPEG 1

MPEG 2

MPEG 4

MPEG 4-AVC (Advanced Video Coding)

解像度（Resolution）：

1920x1080

1280x720

720x480

720x565

アスペクト比 (Aspect)

4 : 3

16 : 9

フレームレート (Framerate)

60

59.94

50

30

29.97

25

24

23.976

下記はオーディオ属性 (Audio) の持つフィールドと夫々が持ち得る値である。

圧縮方式 (Coding) :

AC3

MPEG1

MPEG2

LPCM

チャンネル数 (Ch) :

1 ~ 8

言語属性 (Language) :

タイムマップ (TMAP) はVOBU毎の情報を持つテーブルであって、当該VOBが有するVOBU数 (Number) と各VOBU情報 (VOBU#1～VOBU#n) を持つ。個々のVOBU情報は、VOBUの再生時間長 (Duration) とVOBUのデータサイズ (Size) を夫々有している。

図14はVOBU情報の詳細を説明する図である。

広く知られているように、MPEGビデオストリームは高画質記録するために可変ビットレート圧縮されることがあり、その再生時間とデータサイズ間に単純な相関はない。逆に、音声の圧縮規格であるAC3は固定ビットレートでの圧縮を行っているため、時間とアドレスとの関係は1次式によって求めることができる。しかしながらMPEGビデオデータの場合は、個々のフレームは固定の表示時間、例えばNTSCの場合は1フレームは $1/29.97$ 秒の表示時間を持つが、個々のフレームの圧縮後のデータサイズは絵の特性や圧縮に使ったピクチャタイプ、いわゆるI/P/Bピクチャによってデータサイズは大きく変わってくる。従って、MPEGビデオの場合は、時間とアドレスの関係は一次式の形で表現することは不可能である。

当然の事として、MPEGビデオデータを多重化しているMPEGシステムストリーム、即ちVOBも時間とデータサイズとを一次式の形で表現することは不可能である。このため、VOB内での時間とアドレスとの関係を結びつけるのがタイムマップ (TMAP) である。図14に示すように、各VOBU毎にVOBU内のフレーム数と、VOBU内のパック数 (つまりデータサイズ) を夫々エントリーとして持つテーブルがタイムマップ (TMAP) である。

図15を使って、タイムマップ (TMAP) を詳細に説明する。

図15に示すように時刻情報が与えられた場合、まずは当該時刻がどのVOBUに

属するのかを検索する。これは、タイムマップのVOBU毎のフレーム数を加算して行き、フレーム数の和が当該時刻を（フレーム数に換算して）超えるまたは一致するVOBUが当該VOBUになる。次にタイムマップのVOBU毎のデータサイズを当該VOBUの直前のVOBUまで加算する。この値が与えられた時刻を含むフレームを再生するために読み出すべきパックのアドレスを求めるために用いられる。

次に図16を使って、プレイリスト情報（"XXX.PL"）の内部構造を説明する。プレイリスト情報は、セルリスト（CellList）とイベントリスト（EventList）から構成されている。

セルリスト（CellList）は、プレイリスト内の再生セルシーケンスであり、本リストの記述順でセルが再生される事になる。セルリスト（CellList）の中身は、セルの数（Number）と各セル情報（Cell#1～Cell#n）である。

セル情報（Cell#）は、VOBファイル名（VOBName）、当該VOB内での開始時刻（In）および終了時刻（Out）と、字幕テーブル（SubtitleTable）を持っている。開始時刻（In）および終了時刻（Out）は、夫々当該VOB内でのフレーム番号で表現され、前述したタイムマップ（TMAP）を使うことによって再生に必要なVOBデータのアドレスを得る事ができる。

字幕テーブル（SubtitleTable）は、当該VOBと同期再生される字幕情報を持つテーブルである。字幕は音声同様に複数の言語を持つことができ、字幕テーブル（SubtitleTable）最初の情報も言語数（Number）とそれに続く個々の言語ごとのテーブル（Language#1～Language#k）から構成されている。

各言語のテーブル（Language#）は、言語情報（Lang）と、個々に表示される字幕の字幕情報数（Number）と、個々に表示される字幕の字幕情報（Speech#1～Speech#j）から構成され、字幕情報（Speech#）は対応するイメージデータファイル名（Name）、字幕表示開始時刻（In）および字幕表示終了時刻（Out）と、

字幕の表示位置 (Position) から構成されている。

イベントリスト (EventList) は、当該プレイリスト内で発生するイベントを定義したテーブルである。イベントリストは、イベント数 (Number) に続いて個々のイベント (Event#1~Event#m) から構成され、個々のイベント (Event#) は、イベントの種類 (Type)、イベントのID (ID)、イベント発生時刻 (Time) と有効期間 (Duration) から構成されている。

図17は、個々のプレイリスト毎のイベントハンドラ (時間イベントと、メニュー選択用のユーザイベント) を持つイベントハンドラテーブル ("XXX.PROG") である。

イベントハンドラテーブルは、定義されているイベントハンドラ/プログラム数 (Number) と個々のイベントハンドラ/プログラム (Program#1~Program#n) を有している。各イベントハンドラ/プログラム (Program#) 内の記述は、イベントハンドラ開始の定義 (<event\_handler>タグ) と前述したイベントのIDと対になるイベントハンドラのID (ID) を持ち、その後に当該プログラムもFunctionに続く括弧 " {" と " } " の間に記述する。前述の "XXX.PL" のイベントリスト (EventList) に格納されたイベント (Event#1~Event#m) は "XXX.PROG" のイベントハンドラのID (ID) を用いて特定される。

次に図18を用いてBDディスク全体に関する情報 ("BD.INFO") の内部構造を説明する。

BDディスク全体情報は、タイトルリスト (TitleList) とグローバルイベント用のイベントテーブル (EventList) から構成されている。

タイトルリスト (TitleList) は、ディスク内のタイトル数 (Number) と、これに続く各タイトル情報 (Title#1~Title#n) から構成されている。個々のタイトル情報 (Title#) は、タイトルに含まれるプレイリストのテーブル (PLTable)



とタイトル内のチャプタリスト (ChapterList) を含んでいる。プレイリストのテーブル (PLTable) はタイトル内のプレイリストの数 (Number) と、プレイリスト名 (Name) 即ちプレイリストのファイル名を有している。

チャプタリスト (ChapterList) は、当該タイトルに含まれるチャプタ数 (Number) と個々のチャプタ情報 (Chapter#1~Chapter#n) から構成され、個々のチャプタ情報 (Chapter#) は当該チャプタが含むセルのテーブル (CellTable) を持ち、セルのテーブル (CellTable) はセル数 (Number) と個々のセルのエントリ情報 (CellEntry#1~CellEntry#k) から構成されている。セルのエントリ情報 (CellEntry#) は当該セルを含むプレイリスト名と、プレイリスト内でのセル番号によって記述されている。

イベントリスト (EventList) は、グローバルイベントの数 (Number) と個々のグローバルイベントの情報を持っている。ここで注意すべきは、最初に定義されるグローバルイベントは、ファーストイベント (FirstEvent) と呼ばれ、BDディスクがプレーヤに挿入された時、最初に呼ばれるイベントである。グローバルイベント用イベント情報はイベントタイプ (Type) とイベントのID (ID) だけを持っている。

図19は、グローバルイベントハンドラのプログラムのテーブル ("BD.PROG") である。

本テーブルは、図17で説明したイベントハンドラテーブルと同一内容である。

(イベント発生メカニズム)

図20から図22を使ってイベント発生メカニズムについて説明する。

図20はタイムイベントの例である。

前述したとおり、タイムイベントはプレイリスト情報 ("XXX.PL") のイベントリスト (EventList) で定義される。タイムイベントとして定義されているイ

イベント、即ちイベントタイプ (Type) が "TimeEvent" の場合、イベント生成時刻 ("t1") になった時点で、ID "Ex1" を持つタイムイベントがシナリオプロセッサからプログラムプロセッサに対してあげられる。プログラムプロセッサは、イベントID "Ex1" を持つイベントハンドラを探し、対象のイベントハンドラを実行処理する。例えば、本実施例の場合では、2つのボタンイメージの描画を行うなどを行うことができる。

図21はメニュー操作を行うユーザーイベントの例である。

前述したとおり、メニュー操作を行うユーザーイベントもプレイリスト情報 ("XXX.PL") のイベントリスト (EventList) で定義される。ユーザーイベントとして定義されるイベント、即ちイベントタイプ (Type) が "UserEvent" の場合、イベント生成時刻 ("t1") になった時点で、当該ユーザーイベントがレディとなる。この時、イベント自身は未だ生成されてはいない。当該イベントは、有効期間情報 (Duration) で記される期間レディ状態にある。

図21に描くように、ユーザがリモコンキーの「上」「下」「左」「右」キーまたは「決定」キーを押した場合、先ずUOPイベントがUOPマネージャによって生成されプログラムプロセッサに上げられる。プログラムプロセッサは、シナリオプロセッサに対してUOPイベントを流し、シナリオプロセッサはUOPイベントを受け取った時刻に有効なユーザーイベントが存在するかを検索し、対象となるユーザーイベントがあった場合は、ユーザーイベントを生成し、プログラムプロセッサに持ち上げる。プログラムプロセッサでは、イベントID "Ev1" を持つイベントハンドラを探し、対象のイベントハンドラを実行処理する。例えば、本実施例の場合では、プレイリスト#2の再生を開始する。

生成されるユーザーイベントには、どのリモコンキーがユーザによって押されたかの情報は含まれていない。選択されたリモコンキーの情報は、UOPイベントによってプログラムプロセッサに伝えられ、仮想プレーヤが持つレジスタSPRM (8) に記録保持される。イベントハンドラのプログラムは、このレジスタの値を調べ

分岐処理を実行することが可能である。

図22はグローバルイベントの例である。

前述したとおり、グローバルイベントはBDディスク全体に関する情報（"BD.INFO"）のイベントリスト（EventList）で定義される。グローバルイベントとして定義されるイベント、即ちイベントタイプ（Type）が"GlobalEvent"の場合、ユーザのリモコンキー操作があった場合にのみイベントが生成される。

ユーザが"メニュー"を押した場合、先ずUOPイベントがUOPマネージャによって生成されプログラムプロセッサに上げられる。プログラムプロセッサは、シナリオプロセッサに対してUOPイベントを流し、シナリオプロセッサは、該当するグローバルイベントを生成し、プログラムプロセッサに送る。プログラムプロセッサでは、イベントID"menu"を持つイベントハンドラを探し、対象のイベントハンドラを実行処理する。例えば、本実施例の場合ではプレイリスト#3の再生を開始している。

本実施例では、単に"メニュー"キーと呼んでいるが、DVDのように複数のメニューキーがあってもよい。各メニューキーに対応するIDを夫々定義することで対応することが可能である。

（仮想プレーヤマシン）

図23を用いてプログラムプロセッサの機能構成を説明する。

プログラムプロセッサは、内部に仮想プレーヤマシンを持つ処理モジュールである。仮想プレーヤマシンはBDとして定義された機能モデルであって、各BDプレーヤの実装には依存しないものである。即ち、どのBDプレーヤにおいても同様の機能を実行することができることを保証している。

仮想プレーヤマシンは大きく2つの機能を持っている。プログラミング関数と

プレーヤ変数（レジスタ）である。プログラミング関数は、JavaScriptをベースとして、以下に記す2つの機能をBD固有関数として定義している。

リンク関数：現在の再生を停止し、指定するプレイリスト、セル、時刻からの再生を開始する

Link (PL#, Cell#, time)

PL# : プレイリスト名

Cell# : セル番号

time : セル内での再生開始時刻

PNG描画関数：指定PNGデータをイメージプレーンに描画する

Draw (File, X, Y)

File : PNGファイル名

X : X座標位置

Y : Y座標位置

イメージプレーンクリア関数：イメージプレーンの指定領域をクリアする

Clear (X, Y, W, H)

X : X座標位置

Y : Y座標位置

W : X方向幅

H : Y方向幅

プレーヤ変数は、プレーヤの状態を示すシステムパラメータ（SPRM）と一般用途として使用可能なゼネラルパラメータ（GPRM）とがある。

図24はシステムパラメータ（SPRM）の一覧である。

SPRM (0) : 言語コード

SPRM (1)	:	音声ストリーム番号
SPRM (2)	:	字幕ストリーム番号
SPRM (3)	:	アングル番号
SPRM (4)	:	タイトル番号
SPRM (5)	:	チャプタ番号
SPRM (6)	:	プログラム番号
SPRM (7)	:	セル番号
SPRM (8)	:	選択キー情報
SPRM (9)	:	ナビゲーションタイマー
SPRM (10)	:	再生時刻情報
SPRM (11)	:	カラオケ用ミキシングモード
SPRM (12)	:	パレンタル用国情報
SPRM (13)	:	パレンタルレベル
SPRM (14)	:	プレーヤ設定値 (ビデオ)
SPRM (15)	:	プレーヤ設定値 (オーディオ)
SPRM (16)	:	音声ストリーム用言語コード
SPRM (17)	:	音声ストリーム用言語コード (拡張)
SPRM (18)	:	字幕ストリーム用言語コード
SPRM (19)	:	字幕ストリーム用言語コード (拡張)
SPRM (20)	:	プレーヤリージョンコード
SPRM (21)	:	予備
SPRM (22)	:	予備
SPRM (23)	:	再生状態
SPRM (24)	:	予備
SPRM (25)	:	予備
SPRM (26)	:	予備
SPRM (27)	:	予備
SPRM (28)	:	予備
SPRM (29)	:	予備

SPRM (3 0) : 予備

SPRM (3 1) : 予備

なお、本実施例では、仮想プレーヤのプログラミング関数をJavaScriptベースとしたが、JavaScriptではなく、UNIX OSなどで使われているB-Shellや、Perl Scriptなど他のプログラミング関数であっても構わなく、言い換えれば、本発明はJavaScriptに限定されるものではない。

#### (プログラムの例)

図25および図26は、イベントハンドラでのプログラムの例である。

図25は、2つの選択ボタンを持ったメニューの例である。

セル (PlayList#1.Cell#1) 先頭でタイムイベントを使って図25左側のプログラムが実行される。ここでは、最初にゼネラルパラメータの一つGPRM (0) に” 1” がセットされている。GPRM (0) は、当該プログラムの中で、選択されているボタンを識別するのに使っている。最初の状態では、左側に配置するボタン1が選択されている事を初期値として持たされている。

次に、PNGの描画を描画関数であるDrawを使ってボタン1、ボタン2夫々について行っている。ボタン1は、座標 (10、200) を起点 (左端) としてPNGイメージ” 1 black.png” を描画している。ボタン2は、座標 (330、200) を起点 (左端) としてPNGイメージ” 2 white.png” を描画している。

また、本セル最後ではタイムイベントを使って図25右側のプログラムが実行される。ここでは、Link関数を使って当該セルの先頭から再度再生するように指定している。

図26は、メニュー選択のユーザイベントのイベントハンドラの例である。

「左」キー、「右」キー、「決定」キー何れかのリモコンキーが押された場合

夫々に対応するプログラムがイベントハンドラに書かれている。ユーザがリモコンキーを押した場合、図21で説明したとおり、ユーザイベントが生成され、図26のイベントハンドラが起動されることになる。本イベントハンドラでは、選択ボタンを識別しているGPRM(0)の値と、選択されたリモコンキーを識別するSPRM(8)を使って分岐処理を行っている。

条件1) ボタン1が選択されている、かつ、選択キーが「右」キーの場合  
GPRM(0)を2に再設定して、選択状態にあるボタンを右ボタン2に変更する。

ボタン1、ボタン2のイメージを夫々書き換える。

条件2) 選択キーが「決定(OK)」の場合で、ボタン1が選択されている場合

プレイリスト#2の再生を開始する

条件3) 選択キーが「決定(OK)」の場合で、ボタン2が選択されている場合

プレイリスト#3の再生を開始する

上記のようにして実行処理が行われる。

(プレーヤ処理フロー)

次に図27から図30を用いてプレーヤでの処理フローを説明する。

図27は、AV再生までの基本処理フローである。

BDディスクを挿入すると(S101)、BDプレーヤはBD.INFOファイルの読み込みと解析(S102)、BD.PROGの読み込み(S103)を実行する。BD.INFOおよびBD.PROGは共に管理情報記録メモリに一旦格納され、シナリオプロセッサによって解析される。

続いて、シナリオプロセッサは、BD.INFOファイル内のファーストイベント (FirstEvent) 情報に従い、最初のイベントを生成する (S104)。生成されたファーストイベントは、プログラムプロセッサで受け取られ、当該イベントに対応するイベントハンドラを実行処理する (S105)。

ファーストイベントに対応するイベントハンドラには、最初に再生すべきプレイリスト情報が記録されていることが期待される。仮に、プレイリスト再生が指示されていない場合には、プレーヤは何も再生することなく、ユーザイベントを受け付けるのを待ち続けるだけになる。(S201)。BDプレーヤはユーザからのリモコン操作を受け付けると、UOPマネージャはプログラムマネージャに対してUOPイベントを立ち上げる (S202)。

プログラムマネージャは、UOPイベントがメニューキーかを判別し (S203)、メニューキーの場合は、シナリオプロセッサにUOPイベントを流し、シナリオプロセッサがユーザイベントを生成する (S204)。プログラムプロセッサは生成されたユーザイベントに対応するイベントハンドラを実行処理する (S205)。

図28は、PL再生開始からVOB再生開始までの処理フローである。

前述したように、ファーストイベントハンドラまたはグローバルイベントハンドラによってプレイリスト再生が開始される (S301)。シナリオプロセッサは、再生対象のプレイリスト再生に必要な情報として、プレイリスト情報 "XXX.PL" の読み込みと解析 (S302)、プレイリストに対応するプログラム情報 "XX.PROG" の読み込みを行う (S303)。続いてシナリオプロセッサは、プレイリストに登録されているセル情報に基づいてセルの再生を指示する (S304)。セル再生は、シナリオプロセッサからプレゼンテーションコントローラに対して要求が出さる事を意味し、プレゼンテーションコントローラはAV再生を開始する (S305)。



AV再生の開始（S4 0 1）を開始すると、プレゼンテーションコントローラは再生するセルに対応するVOBの情報ファイル（XXX.VOBI）を読み込みおよび解析をする（S4 0 2）。プレゼンテーションコントローラは、タイムマップを使って再生開始するVOBUとそのアドレスを特定し、ドライブコントローラに読み出しアドレスを指示し、ドライブコントローラは対象となるVOBデータを読み出し（S4 0 3）、VOBデータがデコーダに送られ再生が開始される（S4 0 4）。

VOB再生は、当該VOBの再生区間が終了するまで続けられ（S4 0 5）、終了すると次のセル再生S3 0 4へ移行する。次にセルが無い場合は、再生が停止する（S4 0 6）。

図29は、AV再生開始後からのイベント処理フローである。

BDプレーヤはイベントドリブン型のプレーヤモデルである。プレイリストの再生を開始すると、タイムイベント系、ユーザイベント系、字幕表示系のイベント処理プロセスが夫々起動され、平行してイベント処理を実行するようになる。

S5 0 0系の処理は、タイムイベント系の処理フローである。

プレイリスト再生開始後（S5 0 1）、プレイリスト再生が終了しているかを確認するステップ（S5 0 2）を経て、シナリオプロセッサは、タイムイベント発生時刻になったかを確認する（S5 0 3）。タイムイベント発生時刻になっている場合には、シナリオプロセッサはタイムイベントを生成し（S5 0 4）、プログラムプロセッサがタイムイベントを受け取りイベントハンドラを実行処理する（S5 0 5）。

ステップS5 0 3でタイムイベント発生時刻になっていない場合、または、ステップS5 0 4でイベントハンドラ実行処理後は再度ステップS5 0 2へ戻り、上述した処理を繰り返す。また、ステップS5 0 2でプレイリスト再生が終了した

ことが確認されると、タイムイベント系の処理は強制的に終了する。

S600系の処理は、ユーザイベント系の処理フローである。

プレイリスト再生開始後（S601）、プレイリスト再生終了確認ステップ（S602）を経て、UOP受付確認ステップの処理に移る（S603）。UOPの受付があった場合、UOPマネージャはUOPイベントを生成し（S604）、UOPイベントを受け取ったプログラムプロセッサはUOPイベントがメニューコールであるかを確認し（S605）、メニューコールであった場合は、プログラムプロセッサはシナリオプロセッサにイベントを生成させ（S607）、プログラムプロセッサはイベントハンドラを実行処理する（S608）。

ステップS605でUOPイベントがメニューコールで無いと判断された場合、UOPイベントはカーソルキーまたは「決定」キーによるイベントである事を示している。この場合、現在時刻がユーザイベント有効期間内であることをシナリオプロセッサが判断し（S606）、有効期間内である場合には、シナリオプロセッサがユーザイベントを生成し（S607）、プログラムプロセッサが対象のイベントハンドラを実行処理する（S608）。

ステップS603でUOP受付が無い場合、ステップS606で現在時刻がユーザイベント有効期間に無い場合、または、ステップS608でイベントハンドラ実行処理後は再度ステップS602へ戻り、上述した処理を繰り返す。また、ステップS602でプレイリスト再生が終了したことが確認されると、ユーザイベント系の処理は強制的に終了する。

図30は字幕処理のフローである。

プレイリスト再生開始後（S701）、プレイリスト再生終了確認ステップ（S702）を経て、字幕描画開始時刻確認ステップに移る（S703）。字幕描画開始時刻の場合、シナリオプロセッサはプレゼンテーションコントローラに字幕描画を指示し、プレゼンテーションコントローラはイメージプロセッサに字幕描

画を指示する（S704）。ステップS703で字幕描画開始時刻で無いと判断された場合、字幕表示終了時刻であるかを確認する（S705）。字幕表示終了時刻であると判断された場合は、プレゼンテーションコントローラがイメージプロセッサに字幕消去指示を行い、描画されている字幕をイメージプレーンから消去する（S706）。

字幕描画ステップS704終了後、字幕消去ステップS706終了後、または、字幕表示終了時刻確認ステップS705で当該時刻でないことが判断された場合、ステップS702に戻り、上述した処理を繰り返す。また、ステップS702でプレイリスト再生が終了したことが確認されると、字幕表示系の処理は強制的に終了する。

#### （実施例2）

次に本発明の第2の実施の形態について説明する。

第2の実施の形態は、BDのアプリケーションにおけるハイライト構造（以下、BDハイライトと呼ぶ）に関する内容である。基本的には第1の実施例に基づく内容であり、拡張または異なる部分を中心に説明する。

#### （ハイライト構造）

図31はBDハイライトのアプリケーションイメージを示す図である。

BDハイライトでは、メニュー画面やユーザーにカーソルキーなどによるボタン操作を促す際、画面にボタンイメージを表示する。ユーザーの操作によりカーソルキーによりボタンのフォーカスが移動し、決定キーを押すことにより選択されたボタンのボタンコマンドが実行される。

ハイライトを実現するために必要な情報は、画面上に表示するボタンのイメージデータ（3101）と、カーソルキーなどによるユーザー操作（3102）でどのような制御を行うか決定するボタン制御情報が必要である。

図32はBDハイライトのカプセル化データ構造を示した図である。

カーソルキーの移動や選択された際のボタンコマンドなどの情報をまとめたボタン制御情報（3201）とボタンイメージデータ（3202）はカプセル化されている。ボタン制御情報とボタンイメージデータをカプセル化することにより、ボタンに依存する情報をまとめることが可能である。

#### （ボタンイメージデータ）

ボタンイメージデータは、PNGデータの集まりである。BDハイライトの表示区間において、同時に表示が開始し終了するボタンイメージデータは、ひとまとめとしてデータを連結しカプセル化されている（3203）。PNGデータはPNGデータごとにイメージのカラー情報を保持している。PNGデータがインデックスカラーを用いるタイプであり、カプセル化された複数のPNGデータから参照されるカラー情報テーブル（CLUT：Color Look Up Table）が共通である場合、それぞれのPNGデータがそれぞれCLUTを保持する必要はなく、カプセル化されたPNG共通CLUT（3204）としてカプセルの先頭に記述しておけばよい。

共通のCLUT情報をまとめることにより、それぞれのPNGに重複してCLUT情報を記述する必要がなくなるため、データ量を抑制することができる。また、同時に表示されるボタンイメージをプレーヤのグラフィクスプレーンに描画する際、共通のCLUTを利用できるため、処理が容易になる。

なお、BDハイライトのカプセル化構造に置いて、共通CLUTがあるにもかかわらず、PNGがそれぞれCLUTを持っていた場合、共通のCLUTが優先されて利用されるようにしてもよいし、個々のPNGデータのCLUTを優先して利用してもよい。またどちらを優先するかボタン制御情報3201に記述しておいても良い。また、PNGデータがインデックスカラーではなく、ピクセルごとのRGBを格納している場合は、そちらを優先して表示するようにしてもよい。

#### （ボタン制御情報）

図33はボタン制御情報を示した図である。

ボタン制御情報（図32の3201）は、該メニュー画面全体に関わるページ

属性情報（3301）と、ページに複数配置されるボタンのそれぞれの属性情報であるボタン属性情報（3302）からなっている。

図34はページ属性情報として必要な属性の一覧を示している。

status\_of\_Highlight（3401）は、直前に表示されていたハイライト（メニュー画面）との関係が記述されている。直前のメニュー画面と同一であるか否かを示し、再描画を行うか否かを判断させる。これはストリーム中に同一のボタン制御情報を複数回埋め込むことで早送り、逆スロー再生等の特殊再生時にハイライトを表示しやすくするためである。start\_PTS\_of\_Highlight（3402）およびend\_PTS\_of\_Highlight（3403）は、ハイライトの開始時刻および終了時刻を、ストリーム中の時刻によって指し示している。end\_PTS\_of\_button\_selection（3404）は、ハイライト区間とは別に、ボタン選択可能な終了時刻を記述している。num\_of\_button\_groups（3405）は、ハイライト用のボタンイメージデータがHD画像用とSD画像用のように複数あった場合に、いくつの画面属性が存在するか示している。button\_display\_types（3406）はHD、SD画面属性ごとの情報であり、それぞれの画面属性情報は、button\_display\_type（3407）によって記述されている。これによってそれぞれのボタングループがどの画面用途なのか識別する。button\_offset\_number（3408）は、ボタン番号とユーザーが指定できるボタン番号が同じでない場合のオフセット値を指定する。例えば、オフセット値が10であれば、ユーザーが数字キーにより12を指定した場合、2番目のボタンが選択されたのと等価になる。num\_of\_buttons\_in\_group（3409）は、画面に表示されるボタンの数であり、num\_of\_numerical\_select\_button（3410）はそのうち数字キーにより選択可能なボタンの数である。forcedly\_selected\_button（3411）はハイライト開始時に選択されているボタンを指定し、forcedly\_activated\_buttonはハイライト終了時に強制的に選択されるボタンを指定するために利用する。

図35はボタンごとの属性情報（図33の3303）として必要な属性の一覧を示している。

normal\_button\_filename (3 5 0 1)、selection\_button\_filename (3 5 0 2)、action\_button\_filename (3 5 0 3) は、それぞれ、ボタンにフォーカスが当たっていない状態で表示するイメージデータファイル名、フォーカスが当たっているときに表示するイメージデータファイル名、選択された時に表示するイメージデータファイル名が記述されている。start\_x\_coordinate (3 5 0 4) と end\_x\_coordinate (3 5 0 5) はイメージデータを表示する矩形のX座標の開始点と終了点を示し、start\_y\_coordinate (3 5 0 6) と end\_y\_coordinate (3 5 0 7) はイメージデータを表示する矩形のY座標の開始点と終了点を示す。auto\_action\_mode (3 5 0 8) は、ボタンにフォーカスが当たったときに、強制的に選択状態にするかしないかを示している。upper\_button\_id (3 5 0 9) と lower\_button\_id (3 5 1 0) と left\_button\_id (3 5 1 1) と right\_button\_id (3 5 1 2) は、それぞれ、カーソルキーの上下左右が押された際にフォーカスが移動するボタン番号が記述されている。button\_command (3 5 1 3) はボタンが選択されたときに実行されるコマンドが記述されている。

なお、normal\_button\_filename属性はなくてもよい。

normal\_button\_filename、selection\_button\_filename、action\_button\_filenameで指定されるイメージファイルは、ボタン制御情報に続くボタンイメージデータ内に格納されており、一つのイメージを複数のボタンから参照することも可能である。あるイメージファイル名で指定されるイメージデータは、カプセル化されたBDハイライト内で一意である必要がある。

透明ボタン（ボタンは存在するが、イメージデータは存在しないボタン）は、イメージデータとして透明なイメージデータを選択してもよい。normal\_button\_filename、selection\_button\_filename、action\_button\_filenameでイメージファイル名を指定しなかった場合は、イメージデータは表示されないため、イメージファイルを指定せず透明ボタンを実現してもよい。

(ハイライトの配置)

以下、ボタン制御情報とボタンイメージデータをカプセル化したデータ構造を、ハイライトユニットと呼ぶことにする。

図36はハイライトユニットの配置の様子を示している。

ハイライトユニット(3601)はボタンのイメージデータを含んでおり、イメージデータのデコードに時間がかかるため、ストリーム(3602)中の実際に表示を開始する時刻より前にデータを配置する必要がある。

通常の再生の場合は、再生経路#1(3605)のような経路を通るため、ボタンの表示開始時刻(3603)より前にハイライトユニットをストリーム中より読み込み、ボタンイメージデータをデコードしておき、表示開始時刻が来るとボタンの表示が可能となっている。表示終了時刻(3604)になるとボタンの表示は終了する。

ただし、ユーザー操作により時間指定での飛び込みや、早送りなどの特殊再生、スキップ操作などが行われた際、再生経路#2(3606)のような経路を通ることになる。前述のように表示開始時刻より前に一度だけハイライトユニットを配置した場合、再生経路#2の場合は、ハイライトユニットを読み飛ばしてしまい、ボタン表示区間内でボタンを表示することができなくなってしまう。

図37は飛び込み再生にも対応したハイライトユニットの配置の様子を示した図である。

ハイライトユニットを一度しか配置しない場合、ユーザーのスキップ操作などにより読み飛ばした場合、ボタンは表示されないことになってしまうため、一定間隔でハイライトユニットを配置する。配置の間隔はコンテンツ作成者の意図により決められる。スキップ操作などにより再生開始区間のどこに飛び込んだとしても、なるべく早くボタンを表示したい場合はハイライトユニットを1秒おきのようになるべく密に配置し、飛び込んでもすぐに表示する必要がない場合は、1

0秒おきにハイライトユニットを配置する等して、ハイライトユニットのために必要なストリームの帯域を抑えることも可能である。

最初のハイライトユニット（3701）と同じボタンを表示するそのほかのハイライトユニット（3702）は、BDハイライトのページ属性の一つであるstatus\_of\_Highlight属性を除いては、同じデータである。status\_of\_Highlight属性は以下のような属性を持つ。

status\_of\_Highlight属性

00b：未定義

01b：新しいハイライトユニット

10b：直前のハイライトユニットと同じデータ

11b：ボタンコマンドを除いて、直前のハイライトユニットと同じデータ

最初のハイライト以外の、同じ属性データを持つハイライトユニットをduplicateハイライトユニット（3702）と呼ぶことにする。最初のハイライトユニットには、表示開始時刻に時刻（3703）を、表示終了時刻は時刻（3704）が指定してある。同じくduplicateハイライトユニットも表示開始時刻以降にデータが置かれていたとしても、最初のハイライトユニットと同じ表示開始時刻と表示終了時刻が記述されている。ボタンイメージデータも最初のものと同じデータを保持している。

最初のハイライトユニットは、status\_of\_Highlight=01b（新しいハイライトユニット）として設定されており、それ以降のduplicateハイライトユニットはstatus\_of\_Highlight=10b（直前のハイライトユニットと同じデータ）と設定されている。最初のハイライトユニットから順にストリームを再生していく再生経路#1（3705）の場合、最初のハイライトユニットを読み込むと、status\_of\_Highlight=01b（新しいハイライトユニット）とあるため、イメージデータのデコードなどを行う。次にduplicateハイライトユニットを読み込むとstatus\_of\_Hig



hlight=10b（直前のハイライトユニットと同じデータ）となっているため、すでにイメージデータをデコードし表示済みのボタンであるため、duplicateハイライトユニットの処理は行わない。

ところが、再生経路#2（3706）の場合は、最初のハイライトユニットを読み込みおよび表示していないにもかかわらず、status\_of\_Highlight=10b（直前のハイライトユニットと同じデータ）となっているduplicateハイライトユニットを読み込むため、最初のハイライトユニットを読み込まなかったと判断して、duplicateハイライトユニットを最初のハイライトユニットとしてイメージデータのデコードを行い、表示開始時刻になるかすでに表示開始時刻を過ぎていれば、ボタンイメージデータを表示する。

duplicateハイライトユニットの配置とstatus\_of\_Highlightの活用により、通常再生時には無駄なイメージデータのデコードを抑制し、飛び込み再生時にもボタンイメージデータを表示することが可能となる。

図38は、status\_of\_Highlight=11b（ボタンコマンドを除いて、直前のハイライトユニットと同じデータ）の使い方を説明するための図である。

ハイライトユニット#1（3801）とハイライトユニット#2（3802）は、ボタンの表示区間は違うが連続しているハイライトである。2つのハイライトにおいてボタンイメージデータは同じであり、連続して再生する際にイメージデータの再デコードは必要ない。そのため、ハイライトユニット#2にstatus\_of\_Highlight=11b（ボタンコマンドを除いて、直前のハイライトユニットと同じデータ）を設定しておくことにより、ハイライト区間やボタンコマンドなどのボタン制御情報は再設定を行うが、イメージデータはそのまま利用することができ、冗長なイメージデータのデコードを抑制することが可能となる。イメージデータがPNGファイルなどであり、ソフトウェアデコードなどによりデコードに時間がかかる場合、これらのstatus\_of\_Highlightの使い方は有効である。

なお、直前のハイライトとの比較だけでなく、飛び込みを行った際もPNGデー

タの抑制を行いたい場合は、ハイライトユニットにIDをつけ、status\_of\_Highlightに、『ハイライト番号??のハイライトユニットと同じデータ』というような属性を追加してもよい。同じボタンデータが長時間にわたり続く場合にスキップが多用されるような状況では有効である。

### (実施例3)

次に本発明の第3の実施の形態について説明する。

第3の実施の形態は、BDのアプリケーションにおけるハイライト構造（以下、BDハイライトと呼ぶ）に関する内容である。基本的には第2の実施例に基づく内容であり、拡張または異なる部分を中心に説明する。

実施例2にあげられたBDハイライトの拡張を以下に示す。

### (ページ属性とボタン属性)

実施例2のページ属性にある属性のうち、いくつかを各々のボタンの属性であるボタン属性に移すことを考える。各々のボタンごとに属性を設定することになるため、トータルのデータ量が増え設定は複雑になるが、ボタンに対するより細かい設定が可能となりユーザー利便性があがる。

実施例2のBDハイライト構造では、数字キーによるボタンの選択を設定する場合、連番でしか設定できない。メニューなどで、コンテンツ1と2と3を選択でき、9で親コンテンツに戻るといったような設定は不可能である。そこで、ページ属性のbutton\_offset\_numberとnum\_of\_buttons\_in\_groupとnum\_of\_numerical\_select\_buttonを廃止し、個々のボタン属性として、user\_button\_number属性を追加する。

数字キーによりボタンが選択される場合、user\_button\_numberを比較して選択されたボタンをチェックする。user\_button\_numberが設定されていない場合は、数字キーによりそのボタンが選択されることはない。この方法であれば、ボタンに対する数字キーの割り当ては連番である必要はなく、数字キーを割り当てたくな

いボタンに対してはuser\_button\_numberを設定しなければよい。

同様に、forcedly\_selected\_button属性やforcedly\_activated\_button属性をボタン属性に移動させてもよい。

#### (ボタン効果音)

ユーザー操作によりボタン間をカーソルが移動するときや選択した際に、効果音が鳴るように拡張する。

図39は、効果音のオーディオデータもハイライトユニットにカプセル化した様子を示した図である。

ボタンイメージデータとともにボタン効果音のための効果音データ(3901)もカプセル化する。効果音データを制御するために、ページ属性情報(3902)とボタン属性情報(3903)に、効果音用の属性情報を追加する。

ボタンからフォーカスがはずれた状態に変化したとき、ボタンにフォーカスが当たったとき、ボタンが選択されたときに出される効果音を、normal\_audio属性(3908)、selection\_audio属性(3909)、action\_audio属性(3910)としてボタン属性に設定する。ボタンごとに音を設定することが可能であり、それぞれの属性を設定しなければ、音は鳴らない。

なお、それぞれの効果音は、ページ内で共通として、ページ属性とすることも可能である。

また、ボタンの状態が変化したときではなく、カーソルキーが押された時に効果音を発生させることも可能である。カーソルキーの上下左右に対応させて、upper\_audio属性(3904)、lower\_audio属性(3905)、left\_audio属性(3906)、right\_audio属性(3907)を設定する。

カーソルキーの効果音とボタンの効果音の両方が設定されている場合、カーソル効果音を優先してもよいし、ボタン効果音を優先してもよい。

#### (ハイライトスチル)

ハイライト終了直前にAV再生を停止させる（ナビタイマーなどは動作状態のまま）にさせる構造（以下、ハイライトスチルと呼ぶ）を、BDハイライトに追加する。

図40は、ハイライトスチルの動作を説明するための図である。

ハイライトスチルが設定されたハイライトユニットを再生する場合、ボタン表示終了時刻の直前で、AVの再生が停止する。ボタン表示終了時刻の直前であるためボタンは表示されたままであり、`forcedly_activated_button`属性で指定されたボタンは選択されていない。また、AVの再生のみ停止、ナビタイマーなどは動作状態のため、カーソルキーの移動などは可能となっている。

前述の機能により、ユーザー操作が行われるまでコンテンツの再生を停止するようなアプリケーションが実現可能となる。

#### (実施例4)

次に本発明の第4の実施の形態について説明する。

第4の実施の形態は、BDのアプリケーションにおけるハイライト構造（以下、BDハイライトと呼ぶ）に関する内容である。基本的には第2の実施例に基づく内容であり、拡張または異なる部分を中心に説明する。

実施例2にあげられたBDハイライトの拡張を以下に示す。

#### (BDハイライト内のイメージデータの格納順序)

BDハイライトを表示する際に、HD画面に映し出すような大きな静止画イメージのデコードには時間がかかる。ここでは、この課題を解決するためのBDハイライ

トデータ構造を示す。

図4 1に示したように、BDハイライトが3つのボタン（A、B、C）から構成される場合の最初の状態をAsBnCn（ボタンAが選択され、ボタンBとCはノーマル状態）と表す。これらのボタンはボタンAとボタンC間の遷移を除き、互いに一つ横のボタンへフォーカス移動することができるのであれば、次に起こりうるBDハイライトの状態は、ボタンAが押されアクティベートした場合のAaBnCnか、右矢印でフォーカス移動した場合のAnBsCnである。

同様に2回目のUOPを受けた後は、ボタンBが押されアクティベートしたAnBaCnか、ボタンCが選択されている状態のAnBnCnである。

このようにBDハイライトを構成する各静止画が表示されるタイミングには順番があり、上記例では最初のBDハイライトの表示までに、ボタンCが押されアクティベートしたイメージデータをデコードしておく必要はない。言い換えれば、静止画のデコード処理に時間がかかるため、時間的に先に必要となるデータからデコードしておくことが理想的である。

前述したように、ボタン制御情報にてボタン間の遷移情報（上下左右のボタンID）と、最初に選択されるべきボタン情報（forcedly\_selected\_button）を格納しており、n回目のUOPで初めて表示される可能性があるボタンイメージを特定することが可能である。

上記例の場合、ボタンA、B、Cについて夫々3つの状態を示す計9枚のイメージデータがあるとすれば、以下のようにデコード優先度を順序付けることができる。

0回目（最初）のUOPで初めて表示される可能性のあるイメージデータ

As、Bn、Cn

1 回目のUOPで初めて表示される可能性のあるイメージデータ

Aa、An、Bs

2 回目のUOPで初めて表示される可能性のあるイメージデータ

Ba、Cs

3 回目のUOPで初めて表示される可能性のあるイメージデータ

Ca

従って、デコード優先順位は、

$As=Bn=Cn > Aa=An=Bs > Ba=Cs > Ca$

となる。

この順番に従いイメージデータを格納したBDハイライトのデータ構造を図4 2に示す。各イメージデータの表示（デコード）順番によって、各々のイメージデータを格納しており、デコーダはこの順序でイメージデータをデコードしていくため、デコード遅延の影響を減らすことができ、迅速にメニュー画面（BDハイライト）を表示することができる。

尚、この格納順序はBDハイライトを構成する全てのイメージデータのデコード優先順位を考慮しなくとも、例えば最初に表示されるイメージデータセットとその他のイメージデータセットで2分割されたBDハイライトを構成したり、最初に表示されるイメージデータセットと一回のUOPで表示される可能性のあるイメージデータセットとその他の（2回以上UOPが発生しないと表示される可能性のない）イメージデータセットで3分割されたBDハイライト構成であっても良い。重要なのは、イメージデータをボタン順でかつステータス順（例えば、AnAsAaBnBsBaCnCsCa）のようにデコード順位を無視して格納しないことにある。

（BDハイライトのPESパケット構造）

上記のように、デコード優先順位を考慮して構成されたBDハイライトをPESパケットとしてAVストリーム中に埋め込むことを考える。

図43に示したのは、BDハイライトを格納したPESパケットの構造である。PESパケットはそのヘッダ情報であるPESパケットと、実データを格納している部分のPESペイロードに分かれる。PESヘッダには、PESペイロードに格納されているデータの種別を表すID (stream\_id) があり、先頭アクセスユニットのデコード時刻 (DTS) と再生開始時刻 (PTS) を示すこともできる。ここで言うアクセスユニットとはデコード・再生の単位であって、BDハイライトのような静止画データであれば、一枚一枚のイメージデータを指す。従ってこのようなPESパケットの構造を用いた場合、PNG#1のDTSとPTSのみがPESヘッダに記載され、他のPNGデータに対するDTS、PTSは記載されないことになり、不定となる。

そこで、DTS、PTSが記載されないイメージデータ (PNG#) に対してもDTS、PTSを確定させるために、ボタン制御情報にそれぞれの値 (DTS#、PTS#) や、DTS#1、PTS#1からの差分情報等を格納し特定させる手段を講じて良い。

また、n回目のUOPで初めて表示される可能性があるイメージデータ単位でPESパケットに格納するようにしても良い。

また、一つの静止画データを一つのPESパケットに格納するようにしても良い。

図44に示したように、UOPの回数に応じてデコード時刻 (再生時刻) が遅延できるブロック (イメージデータのセット) に分けて、PESパケットに格納すると、UOPの回数で表示可能性のあるイメージデータがグループ化されて、指定のDTSのタイミングでデコードされ、PTSのタイミングで表示可能となる。尚、同一ブロック (つまり同一PESパケット) にグループ化されたイメージデータは同じデコード/再生タイミング (DTS/PTS) を持つ。

尚、 $n+1$ 回目のUOPで初めて再生される可能性のあるイメージデータは、 $n$ 回目のUOPで初めて再生される可能性のあるイメージデータに比べて、そのデコード時刻（DTS）と再生時刻（PTS）に遅延があり、尚且つその遅延時間が背景で再生されているビデオストリームの1フレームの再生時間である $T_v$ （ $T_v$ は90 KHzの時間精度で、NTSCだと3003、PALだと3600、Film素材だと3750である。）であっても良いし、 $T_v$ の整数倍であっても良い。または、ビデオのフレームレートとは無関係に、固定時間長であっても良い。

尚、一つのBDハイライトを構成するために、 $n$ 回目のUOPで初めて表示される可能性があるイメージデータごとにPESパケットを割り振るのではなく、最初に表示されるBDハイライトのイメージデータセットを格納したPESパケットと、その他のイメージデータを全て格納したPESパケットの2つに分散配置しても良い。同様に、UOPの回数（デコード優先順位）に応じて3つ以上のPESパケットにBDハイライトのイメージデータを分散配置しても良い。

図45にイメージデータ（PNG）のデコーダモデルを詳細に示した。ここでは、MPEG-TSにて規定されるシステムデコーダモデル（T-STD）を拡張したモデルを例にして説明する。

BDハイライトのPESパケットは、個々のTSパケットに格納されデコーダへ転送されるが、ストリームの種別に応じて各デコーダへ供給される。

#### （MPEGビデオストリームのデコーダモデル）

MPEGビデオストリームの場合は、トランスポートバッファ（TB）4501からRXのレートでマルチプレクシングバッファ（MB）4502へ転送され、Rbxのレートでエレメンタリバッファ（EB）4503へ転送される。

デコードされるデータは、一旦EB4503に蓄積された後、DTSの時刻にビデオデコーダ（Video Dec.）4504に瞬時に引き抜かれ、デコードされ、ビデオ



プレーン (V Plane) 4 5 0 6もしくは、リオーダバッファ (0) 4 5 0 5に展開されたイメージデータが瞬時に (DTSの時刻で) 転送される。

ビデオデータはその再生開始時刻であるPTSの時刻にビデオプレーン (V Plane) 4 5 0 6に展開されており、それが他の映像信号と合成されTV等の表示装置へと出力される。

(字幕用PNGのデコーダモデル)

字幕用PNGストリームの場合は、トランスポートバッファ (TB) 4 5 0 7からRs 1のレートでPNGバッファ (PNG buffer) 4 5 0 8へ転送される。

デコードされるデータは、一旦PNG buffer 4 5 0 8に蓄積された後、DTSの時刻にPNGデコーダ (PNG Decoder) 4 5 0 9に瞬時に引き抜かれ、デコードされ、DIMバッファ (DIM Buffer) 4 5 1 0へ展開されたイメージデータが瞬時に (DTSの時刻で) 転送される。

DIMコントローラ (DIM Controller) 4 5 1 1により、一旦DIMバッファ4 5 1 0に展開されたイメージデータは、指定されたPTSの時間までに字幕プレーン (S Plane) 4 5 1 2の指定された描画領域を示すメモリ空間へレートRs 2でもって転送される。

PTSの時刻に、CLUT 4 5 1 3により色変換された字幕のイメージデータが、他の映像信号と合成されTV等の表示装置へと出力される。

前述のようにPNGのデコードには時間がかかるため、MPEGビデオのデコーダモデルと合致した (デコード時間が一瞬の) デコーダモデルを導入するためには、DTSの設定が問題となる。ここで示したレートRs 1は実際のPNGデコーダがデコードできる処理速度と同じもしくはそれ以下であれば良く、モデルとしては前述のように一瞬でデコードされるが、実機の処理としては、PNGバッファ 4 5 0 8に

転送されたデータはPNGデコーダ4509に逐次転送され、逐次DIMバッファ4510へデコード遅延の後、展開されていく。

また、このモデルは、DIMバッファ4510から字幕プレーン4512への転送がレートRs2と規定されたモデルであるが、このレートRs2はDIMコントローラ4511により指定されたイメージデータをDIMバッファ4510から字幕プレーン4512の該当領域へ展開するのにかかる時間と考えることができる。実際には、バス幅に依存するパラメータである。

(メニュー用BDハイライトのデコーダモデル)

メニュー用BDハイライト用PNGストリームの場合は、トランスポートバッファ(TB)4514からRm1のレートでPNGバッファ(PNG buffer)4515へ転送される。

デコードされるデータは、一旦PNG buffer4515に蓄積された後、DTSの時刻にPNGデコーダ(PNG Decoder)4516に瞬時に引き抜かれ、デコードされ、DIMバッファ(DIM Buffer)4517へ展開されたイメージデータが瞬時に(DTSの時刻で)転送される。

DIMコントローラ(DIM Controller)4518により、一旦DIMバッファ4517に展開されたイメージデータは、指定されたPTSの時間までにメニュープレーン(M Plane)4519の指定された描画領域を示すメモリ空間へレートRm2をもって転送される。

PTSの時刻に、CLUT4520により色変換されたBDハイライトのイメージデータが、他の映像信号と合成されTV等の表示装置へと出力される。

前述のようにPNGのデコードには時間がかかるため、MPEGビデオのデコーダモデルと合致した(デコード時間が一瞬の)デコーダモデルを導入するためには、

DTSの設定が問題となる。ここで示したレートRm1は実際のPNGデコーダがデコードできる処理速度と同じもしくはそれ以下であれば良く、モデルとしては前述のように一瞬でデコードされるが、実機の処理としては、PNGバッファ4515に転送されたデータはPNGデコーダ4516に逐次転送され、逐次DIMバッファ4517へデコード遅延の後、展開されていく。

また、このモデルは、DIMバッファ4517からメニュープレーン4519への転送がレートRm2と規定されたモデルであるが、このレートRm2はDIMコントローラ4518により指定されたイメージデータをDIMバッファ4517からメニュープレーン4519の該当領域へ展開するのにかかる時間と考えることができる。実際には、バス幅に依存するパラメータである。

このようにして、静止画イメージを含むAVストリームの同期再生が可能となる。

#### (字幕用BDハイライト)

これまで特に、メニュー用BDハイライトについて説明してきたが、BDハイライトにはもう一つの用途があり、それは字幕データを格納する場合である。以下メニュー用BDハイライトの制御情報をMCS、字幕用BDハイライトの制御情報をPCSとする。図46に字幕用BDハイライトの構造を示す。

字幕用BDハイライトは、先頭に字幕用のBDハイライトの識別情報(PCS)を持つため、識別が可能である。BDハイライト全体の静止画データの表示制御に関する情報もPCSに格納している。字幕用BDハイライトでは、図に示すように全ての静止画データに関する情報を格納していなくとも良いが、メニュー用BDハイライトではBDハイライト内の全ての静止画データがMCSで管理されている。

図46の例では、字幕データA、B、Cが格納された字幕用BDハイライト#nと、字幕データDが格納されたストリーム中で後続の字幕用BDハイライト#n+1があり、字幕用BDハイライト#nの種別情報PCSには、これが字幕用BDハイライトであることを示す識別情報があり、PCSには、字幕データA、Bの特定可能な識別番号と、夫々の画面上での表示位置情報と、表示終了時刻情報が記述されている。

字幕データA、Bは、字幕用BDハイライト#nのPCSによって規定される字幕データであるが、字幕データCに関する規定は同PCSではなされない。

字幕データCの表示制御は後続の字幕用BDハイライト#n+1のPCSによって行われ、そのPCSに規定される字幕データDと共に表示制御される。これは個々の静止画データが持つ識別番号を利用して実現され、字幕データCは復号化処理だけが字幕用BDハイライト#nの静止画データと同時期に行われる。このように前もって静止画データを送り復号処理を済ませておくことで、字幕BDハイライト#n+1の制御情報（PCS）をデコードし表示する静止画データが確定した時点で、すぐに字幕プレーン（S Plane）4 5 1 2へ表示することが可能となる。また字幕データを識別する識別情報は、ある区間で重複しないことが必要である。例えば字幕BDハイライト#s～#tの区間では重複しないことをAVデータ（1 0 3）やBD管理データ（1 0 2）に記述したり、AVファイル単位（YYY.VOB）では重複しないことを制約しておくことが必要である。

#### （字幕用BDハイライトの静止画データ格納順序）

字幕用BDハイライトの静止画データの格納（多重化）順序は、デコード処理の順番となるため（先にデコード送られた静止画データからデコードされるため）デコードの実装を容易にするために、該字幕用BDハイライトのPCSで表示制御される静止画データと、それ以外の（後続の字幕用BDハイライトのPCSでのみ表示制御される）静止画データの2つに大別することが効果的である。

このように格納することで、表示が早い静止画データを先にデコードしておくことが可能となる。

図4 6を例にして説明すると、字幕データA（PNG#A）と、字幕データB（PNG#B）は字幕用BDハイライト#nのPCSによって表示制御されるデータであるため、後続の字幕用BDハイライト#n+1のPCSによって表示制御される字幕データC（PNG#C）よりも早い表示開始時刻を持つ。そのためデコード処理を早く済ませておくことが求められ、該PCSによって参照されない静止画データよりも前方に格納される。

以下、図4 7を用いて、字幕用静止画データのBDハイライト内での格納順序の詳細について説明する。

図に示したように字幕用BDハイライト内の静止画データは、該PCSによって表示制御される静止画データをグループ化し前方配置し、該PCSによって表示制御されない静止画データをグループ化し後方配置する。

BDハイライト内でPCSによって参照される静止画データは、そのグループの中でスキャンライン順で格納され、参照されない静止画データは、そのグループの中で識別番号順に格納される。

もしくは、BDハイライト内でPCSによって参照される静止画データは、そのグループの中で識別番号順に格納され、参照されない静止画データは、そのグループの中で識別番号順に格納されても良い。

スキャンライン順とは、図示したように静止画データの表示開始座標 ( $x$ ,  $y$ ) を基に、 $y$ 座標の小さい順から並べ、 $y$ 座標が同一の静止画データ間には、 $x$ 座標が小さい順から並ぶことを意味している。

このようにスキャンライン順に並べる方法は、デコーダ実装で1ラインごとの出力処理が容易になることが期待でき、また静止画データの取り扱いが容易であることを意図している。

尚、PCSで参照される静止画データを識別番号順に並べる方法も、デコーダでの静止画データの取り扱いが容易になることを意図している。

#### (メニュー用BDハイライトの静止画データの格納順序)

メニュー用BDハイライトの静止画データの格納(多重化)順序も、字幕の時と同様にデコード処理の順番となるためデコーダの実装を容易にすることと、メニュー表示までのデコード遅延を軽減するために図48に示すような格納順序が考えられる。

図48に示すように、メニューを構成する各ボタンは、実装容易性や、メニュー反応時間の短縮等を目的として、以下のような格納方法が考えられる。

構成を単純化するために、ノーマル(通常)、セレクト(選択時)、アクション(決定時)のイメージデータの状態の違いによってグループ化されても良いし、最初を選択されるのは、ノーマルとセレクトのボタンイメージであることが予想されるため、ノーマルとセレクトのイメージデータを前方にグループ化して、

アクションのイメージデータを後方にグループ化しても良いし、メニューのデコードから表示までを最短化するために、最初に表示される全てのボタンイメージを前方にグループ化し、残りのボタンイメージを後方にグループ化しても良いし、

最初のメニューの状態から、ユーザーが一回の操作をした際に選択され得るメニューの状態までを考慮して、最初に表示される全てのボタンイメージを前方にグループ化し、ユーザーからの一回の操作で表示され得るボタンイメージを次にグループ化し、最後にその他のボタンイメージをグループ化しても良い。

また、各グループの内部では、デコード実装容易性を考慮して、以下のような格納方法が考えられる。

スキャンライン順で配置し、同スキャンライン順の静止画データは識別番号順に格納する、もしくは、グループ内は識別番号順で格納する。

このように規定することで、メニューを構成するあるボタンのある状態が複数枚の静止画イメージから構成される（GIFアニメーションに類似）場合でも、各静止画イメージに付与された識別番号で順番に格納することが可能となる。

尚、上記説明では、静止画データの符号化方法をPNGとして説明したが、これに限らずGIFやRLE（ランレングス圧縮）等の他の符号化方式を用いても同様である。

Although the present invention has been fully described by way of example with reference to accompanying drawing, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Therefore, unless otherwise such changes and modifications depart from scope of the present invention, they should be constructed as being include therein.

【What is claimed is:】

1. 少なくとも一つ以上の静止画データが記録された情報記録媒体であって、  
前記静止画データは、一つ若しくは複数の静止画データを格納する論理ブロックに格納され、  
前記論理ブロック先頭には論理ブロックの制御情報が配され、  
前記制御情報には、  
当該論理ブロックのタイプを示す識別情報と、  
当該論理ブロック内に格納された静止画データの表示管理情報とが配置され、  
前記識別情報が第一の論理ブロックタイプを示している場合、表示管理情報は、  
該論理ブロック内に格納される全ての静止画データの表示位置を示し、  
前記識別情報が第二の論理ブロックタイプを示している場合、表示管理情報は、  
該論理ブロック内に格納される全て、若しくは、一部の静止画データの表示位置を示す。
2. Clam1において前記識別情報が第一の論理ブロックタイプを示している場合、  
識別情報を含む制御情報には、該論理ブロックに格納される全ての静止画データに対する識別番号が割り振られる。
3. Clam2において前記識別情報が第一の論理ブロックタイプを示している場合、  
当該論理ブロックには、  
第一の静止画データグループが先頭から配置される；  
第一の静止画データグループは、ユーザーからの操作が無くとも表示される静止画データからなる。
4. Clam3の前記第一の静止画データグループにおいて、静止画データは、  
前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って配列されている。

5. Clam4の第1の静止画データグループにおいて、表示位置が同一であって、スキャンライン順が同順となる静止画データは、前記識別番号の小さい順に先頭から配置される。

6. Clam3の前記第一の論理グループにおいて、第一の静止画データグループには、

前記第一の論理ブロックに格納される静止画データであって、次に一回のユーザーからの操作に応じて表示される可能性のある静止画データからなる第二の静止画データグループが、後続している。

7. Clam6の前記第2の静止画データグループにおいて、静止画データは、前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って配列されている。

8. Clam7の第2の静止画データグループにおいて、表示位置が同一であって、スキャンライン順が同順となる静止画データは、前記識別番号の小さい順に先頭から配置される。

9. Clam6において前記識別情報が第一の論理ブロックタイプを示している場合、

その第一の論理ブロックに格納される静止画データであって、前記第一の静止画データグループおよび前記第二の静止画データグループに属さない第三の静止画データグループの静止画データが、前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って先頭から配置される。

10. Clam9の第三の静止画データグループにおいて、表示位置が同一であり、スキャンライン順が同順となる静止画データは、前記識別番号の小さい順に先頭から配置される。



1 1. Clia6の第二の静止画データグループに属する静止画データは、前記制御情報に記述された前記識別番号の小さい順に配置されている。

1 2. Clia6の前記第三の静止画データグループ内の静止画データは、前記制御情報に記述された前記識別番号の小さい順に先頭から配置されている。

1 3. Clia1において、前記識別情報が第二の論理ブロックタイプを示している場合、その論理ブロックの制御情報には、該論理ブロックに格納される全てもしくは一部の静止画データに対する識別番号が割り振られている。

1 4. Clia1において、前記識別情報が第二の論理ブロックタイプを示している場合、その論理ブロックには、

前記識別番号が割り振られた全ての静止画データから構成される第四の静止画データグループが、先頭から配置されている。

1 5. Clia1 4の前記第四の静止画データグループに格納される静止画データは、前記制御情報に記述された表示位置情報を基に、スキャンライン順に従って先頭から配置されている。

1 6. Clia1 4において、前記識別情報が第二の論理ブロックタイプを示している場合、その論理ブロックには、

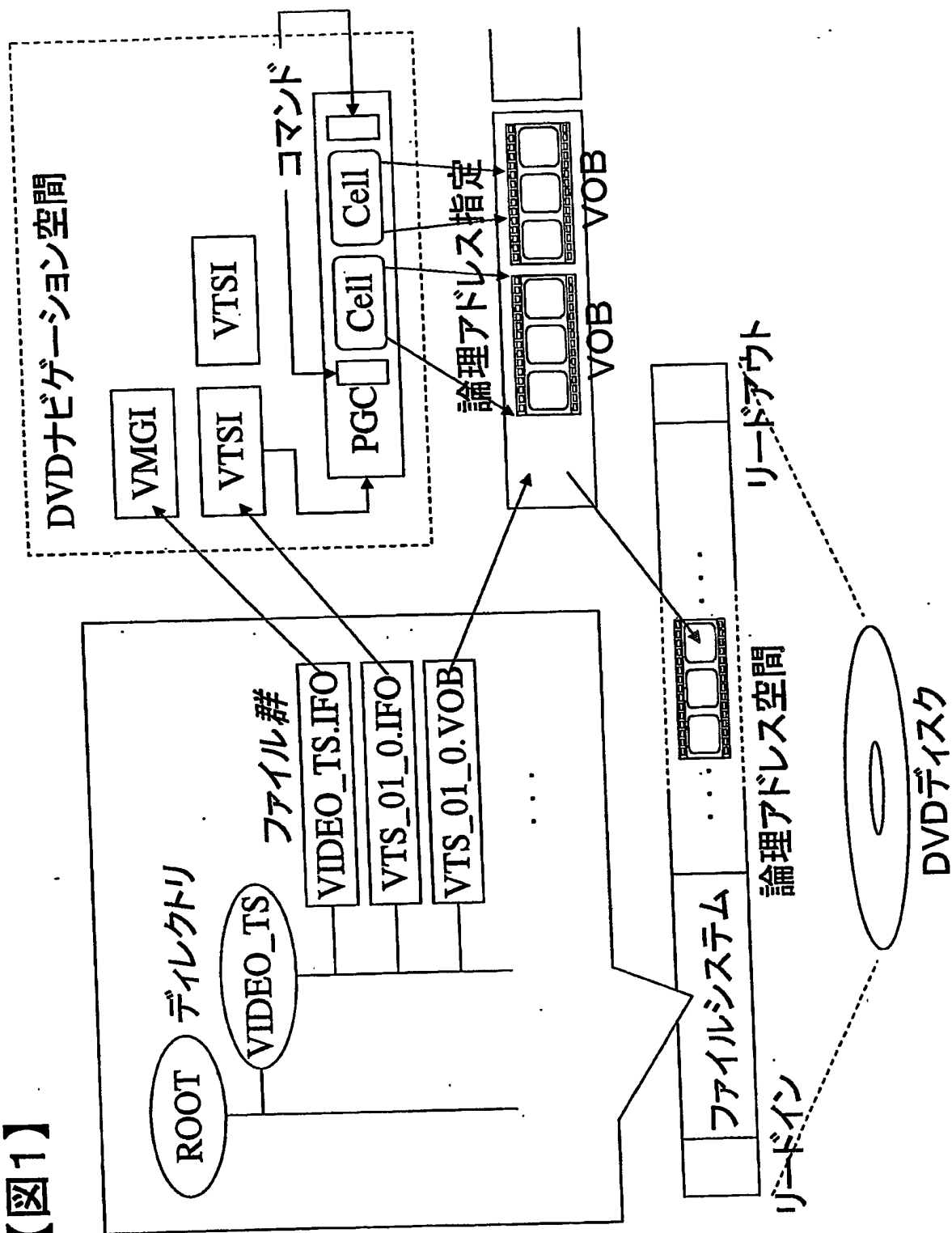
前記識別番号が割り振られていない全ての静止画データから構成される第五の静止画データグループが、前記第四の静止画データグループに続いて配置されている。

【ABSTRACT OF DISCLOSURE.

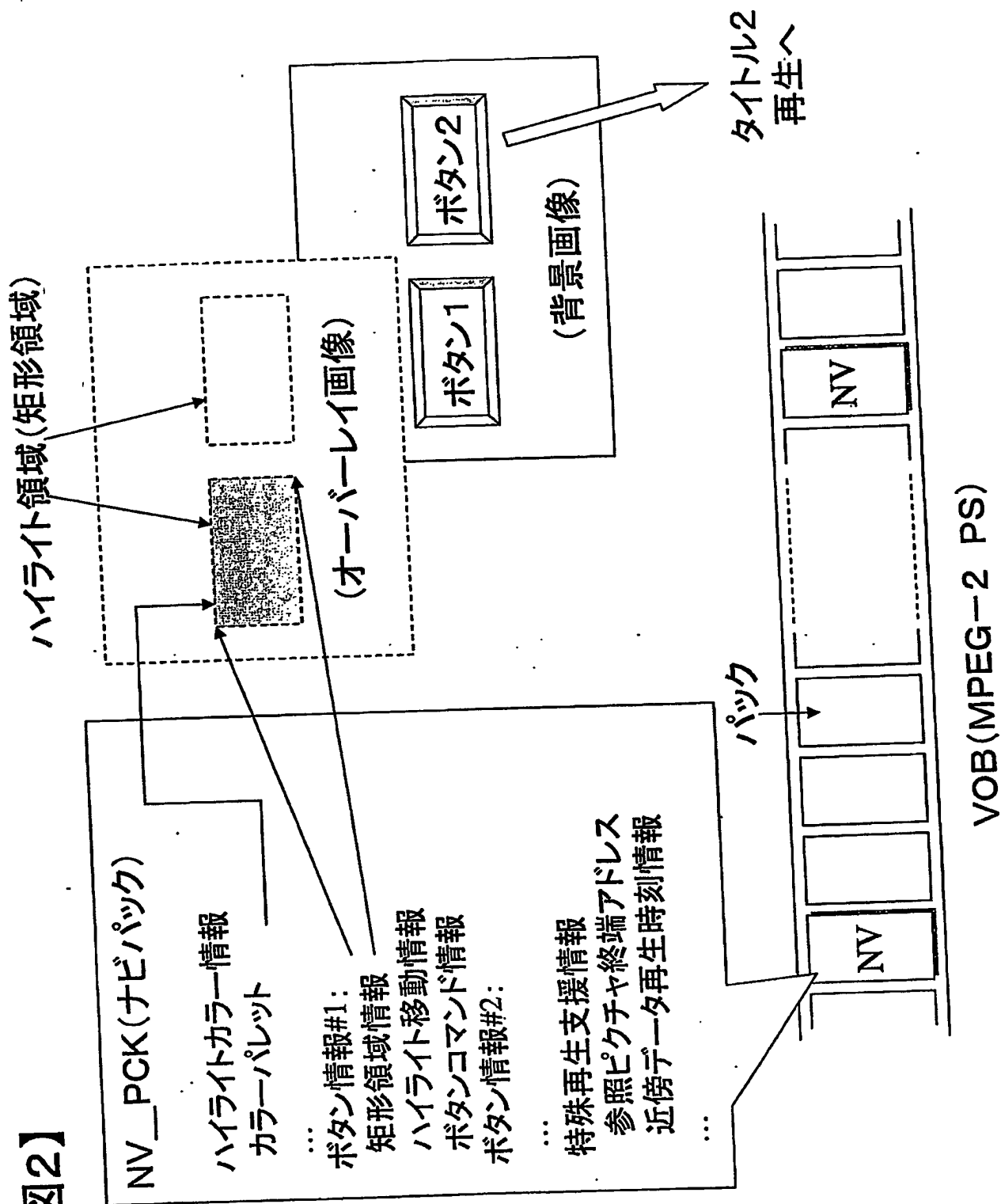
DVDのメニュー画面におけるボタンは、背景にボタン画像を埋め込み、ハイライト部分の色だけを変えて実現していた。また、ハイライト部分の画像データと、ユーザー操作によってどのようにハイライトが変化するか記述したボタン制御情報は分離しており、非常に扱いにくい構造であった。

本発明は、メニュー画面などにおけるボタンイメージとその制御情報をカプセル化している。このため、ハイライトおよびボタンに関するデータを簡単に扱うことが可能である。また、カプセル化の際にデコード遅延を考慮したデコードモデルを規定し、イメージデータのデコード優先度を考慮した多重化を用いることで、迅速にメニューを表示することが可能となる。

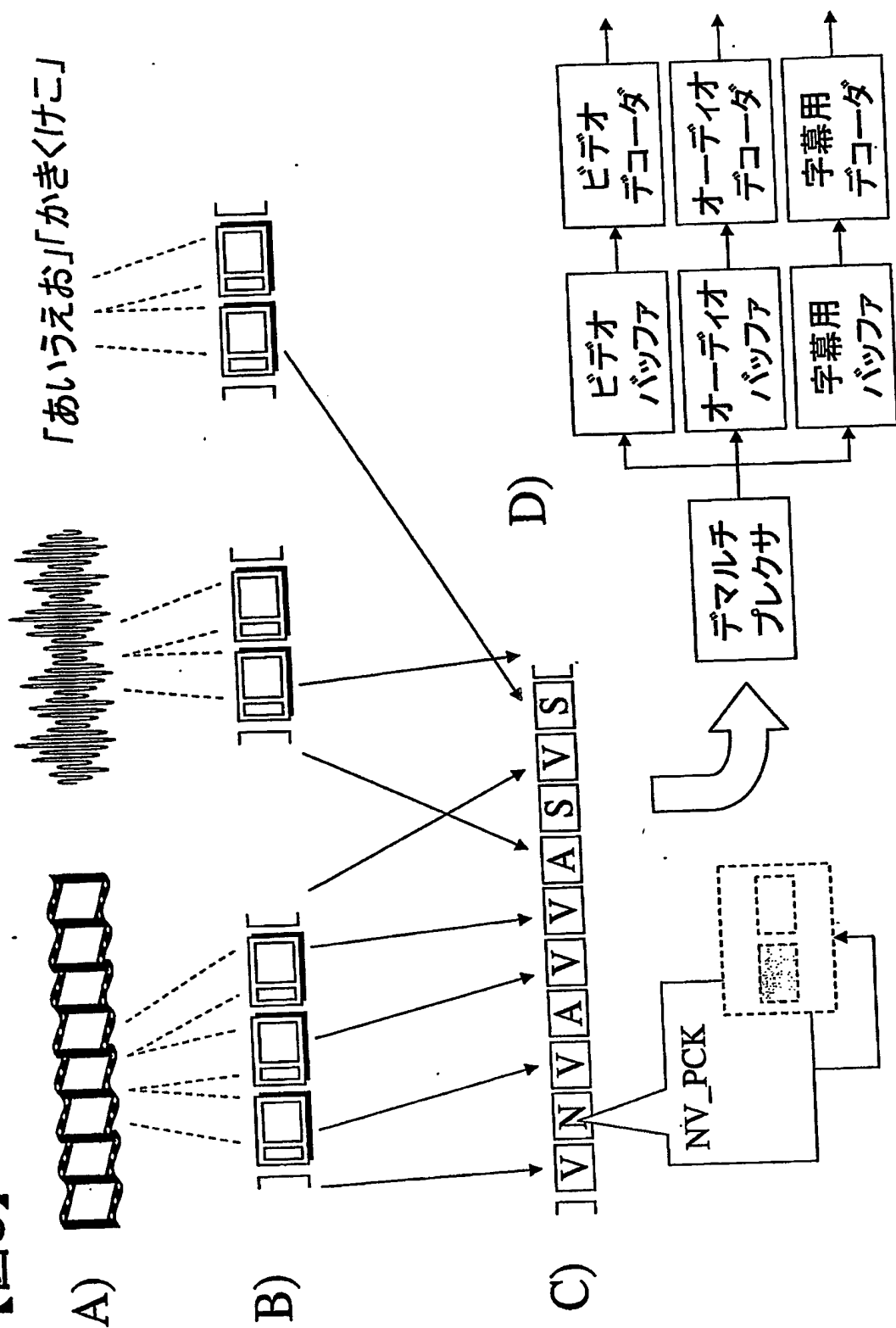
【図1】



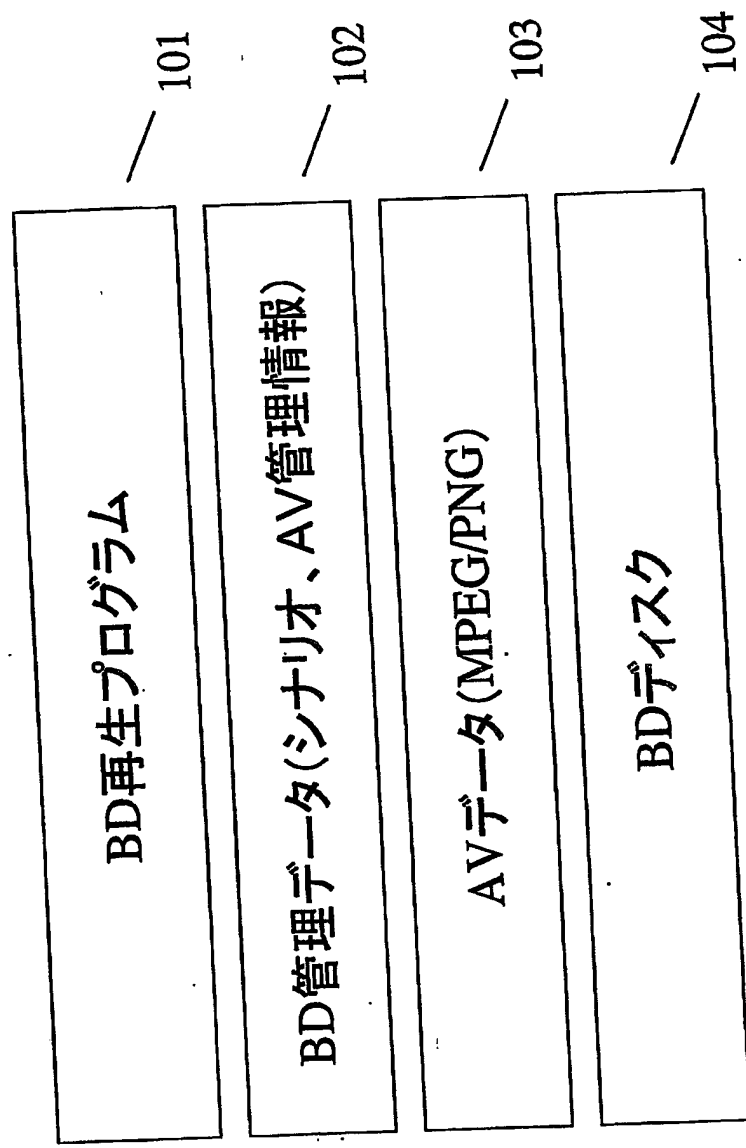
【図2】



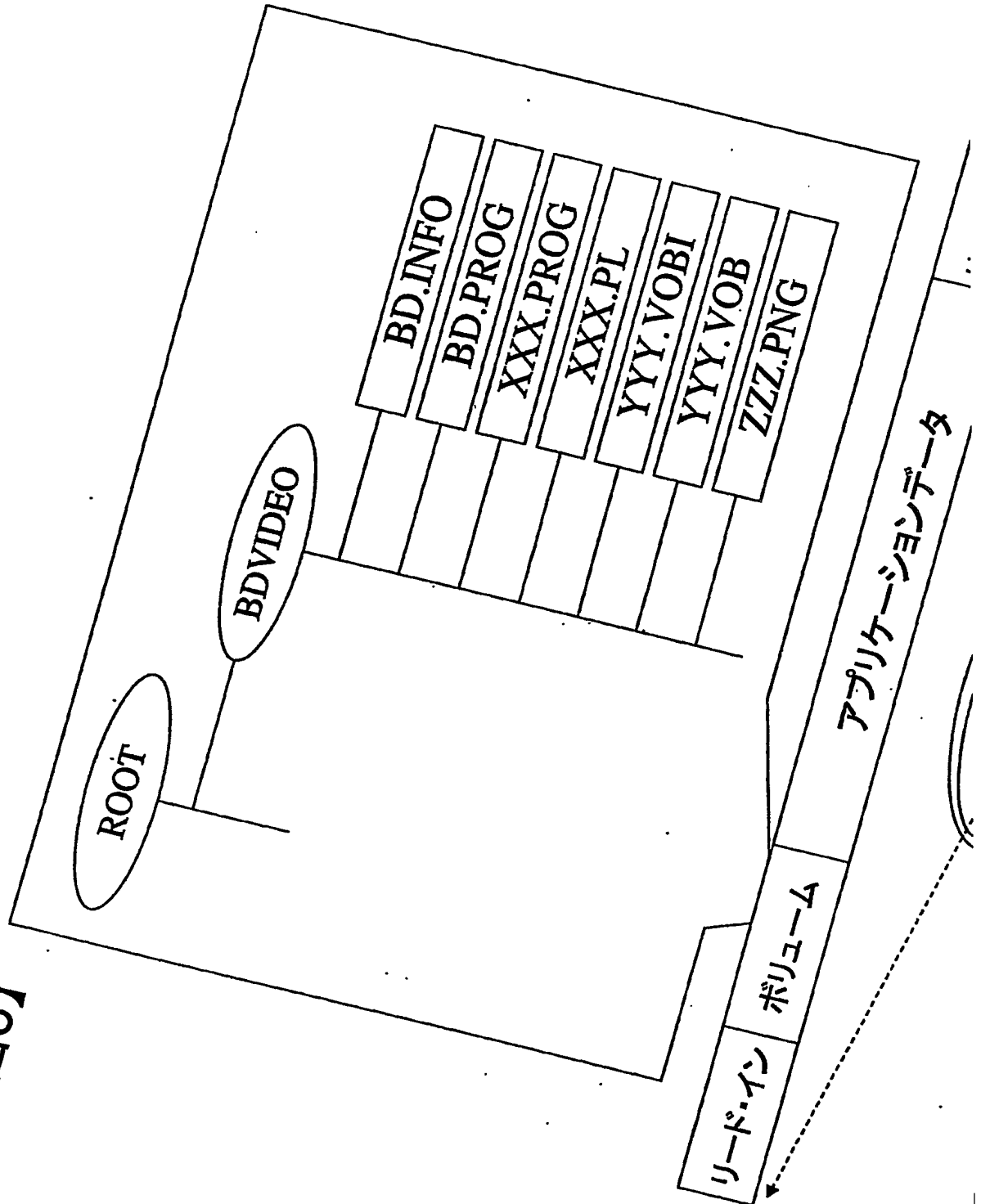
【図3】



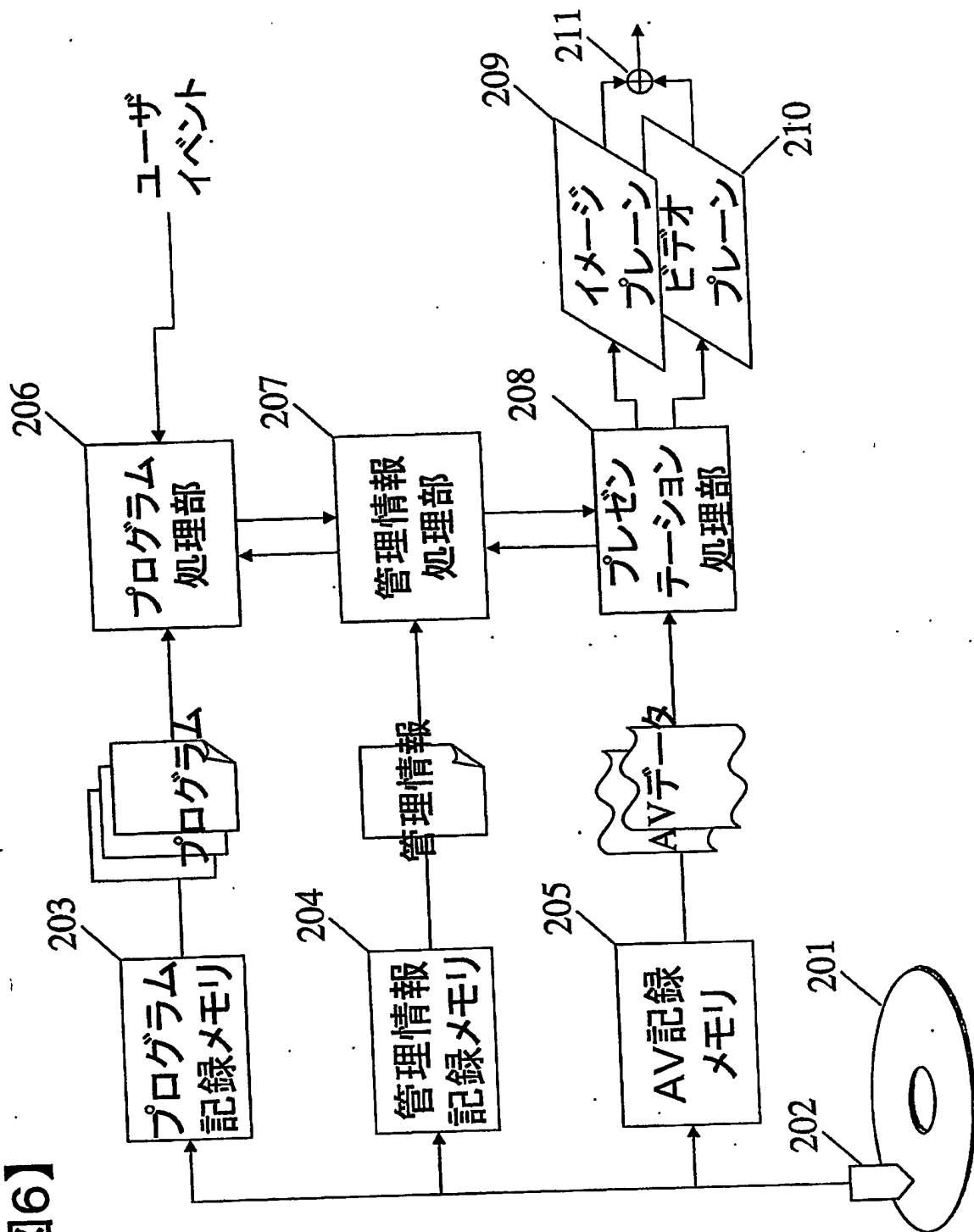
【図4】



【図5】

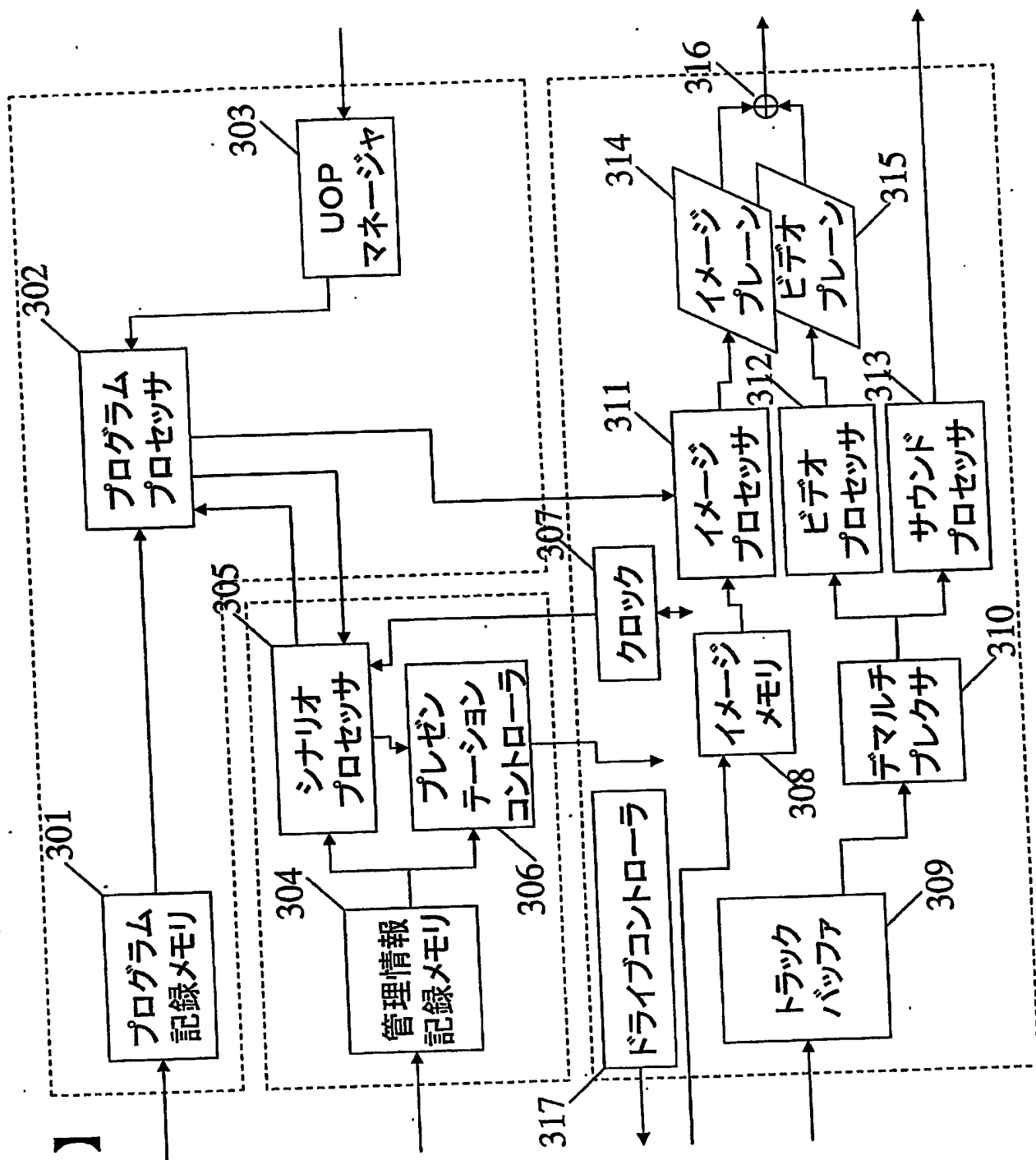


【図6】

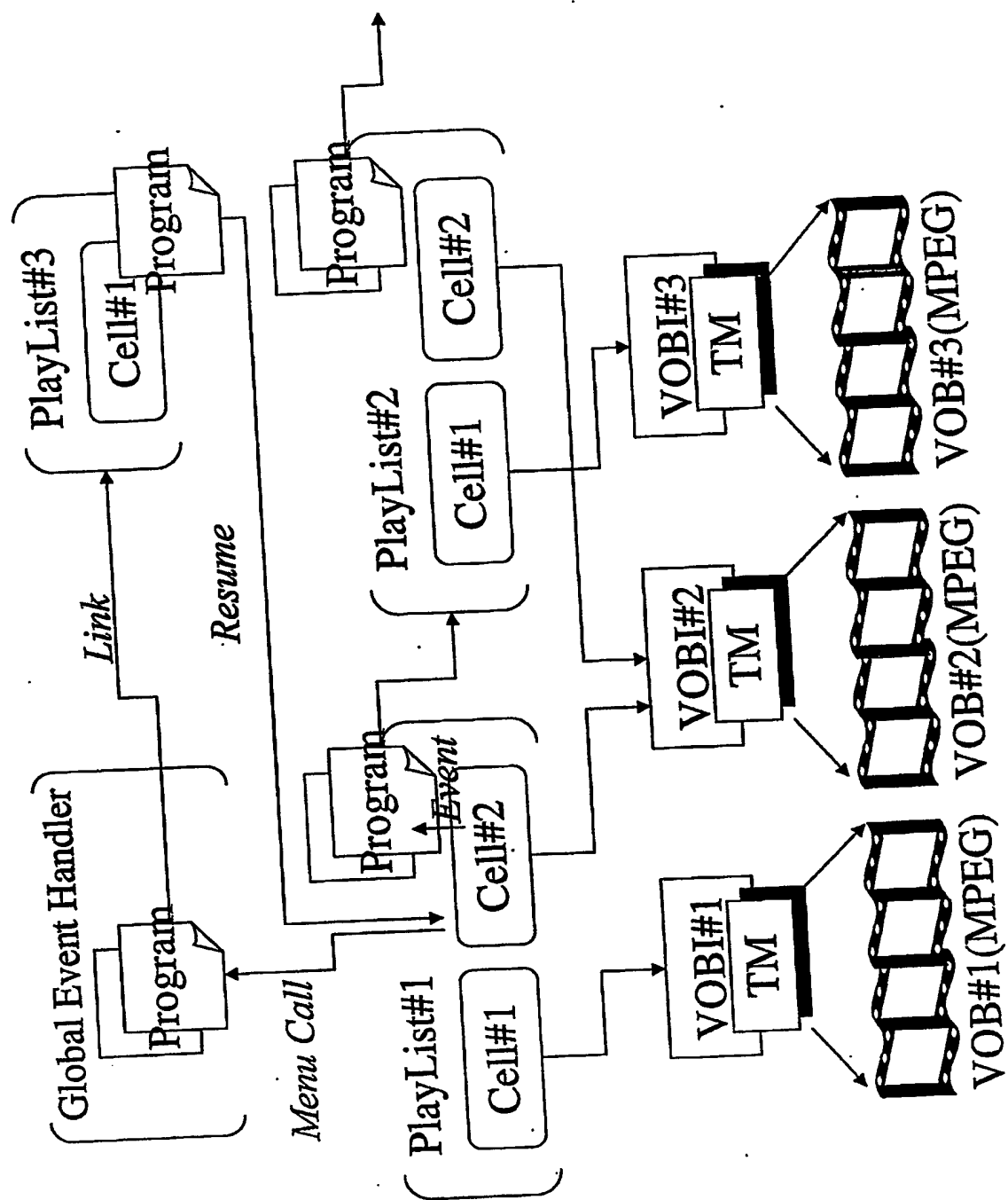




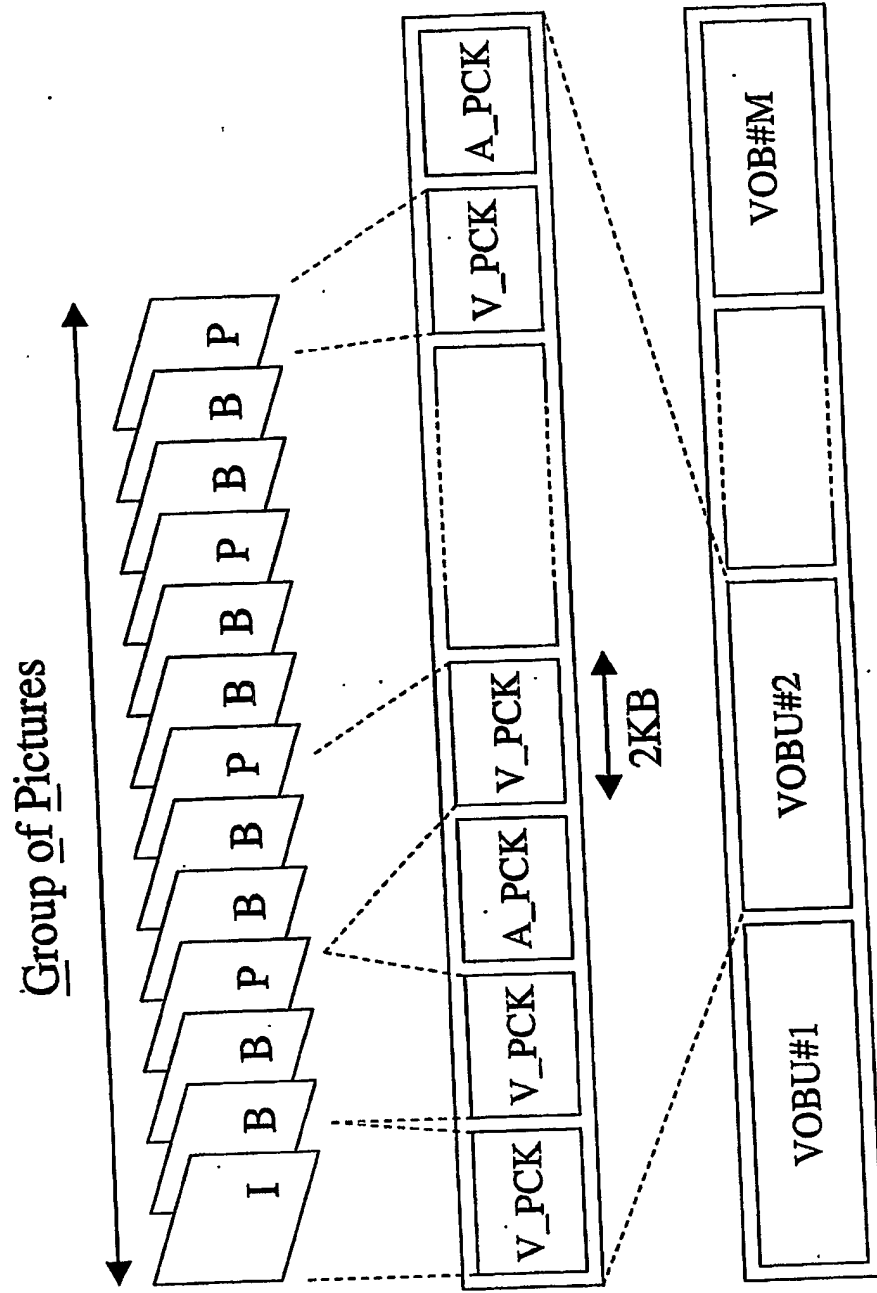
【図7】



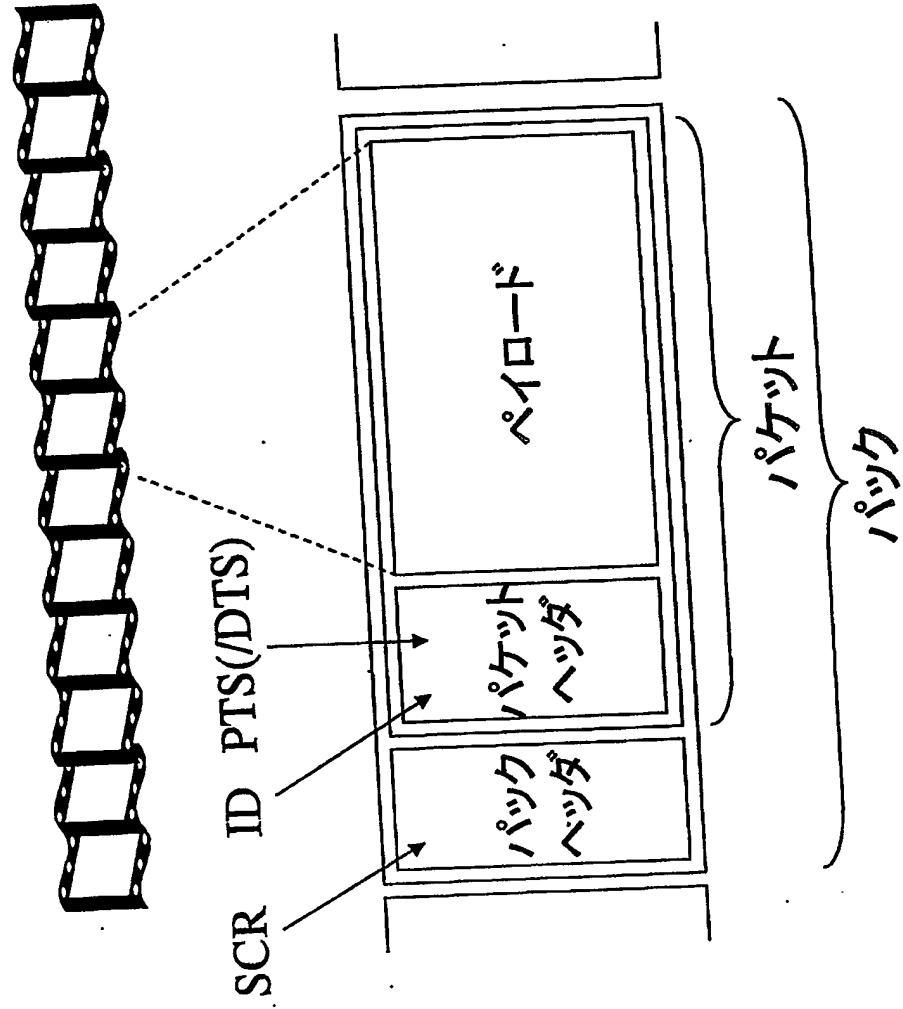
【図8】



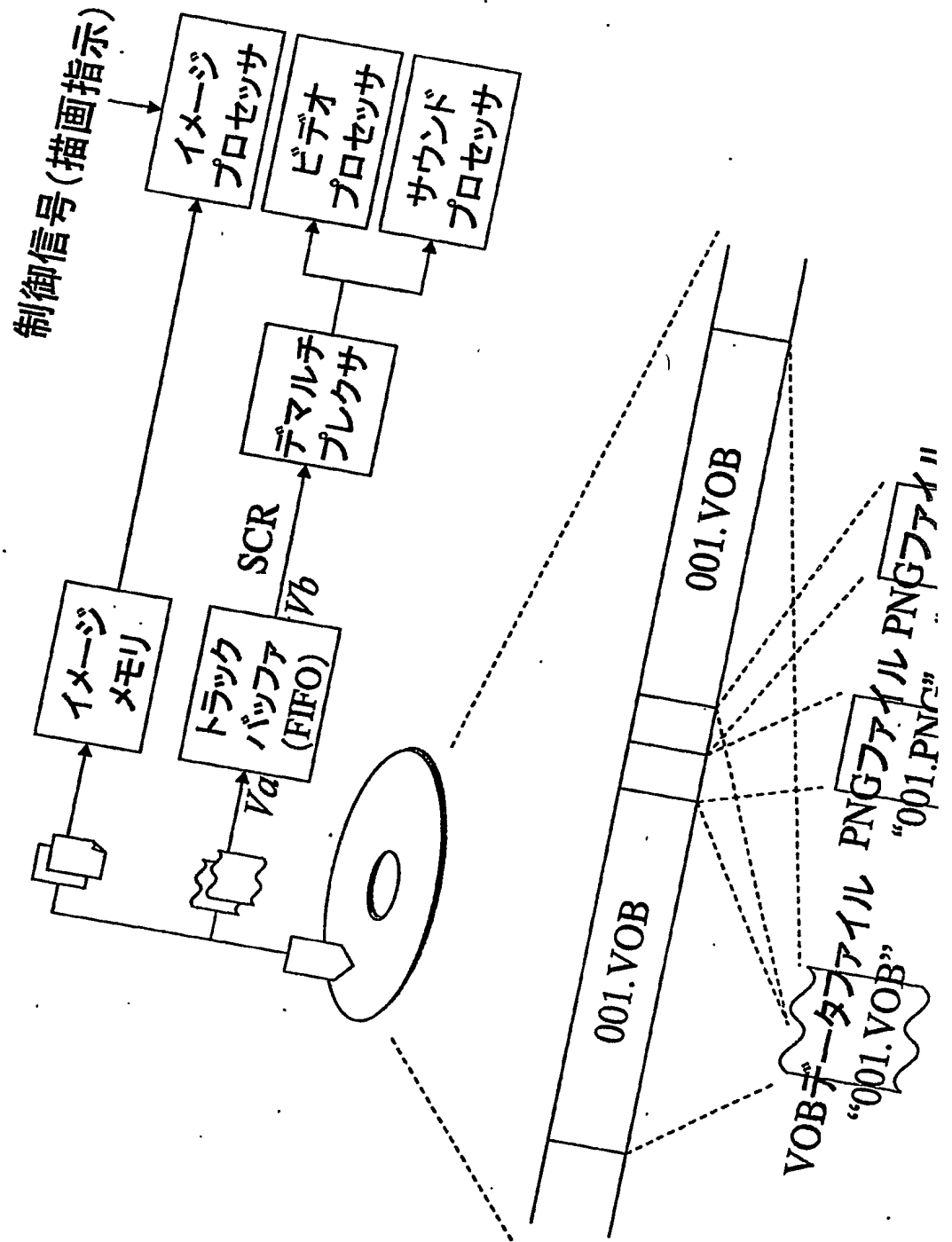
【図9】



【図10】

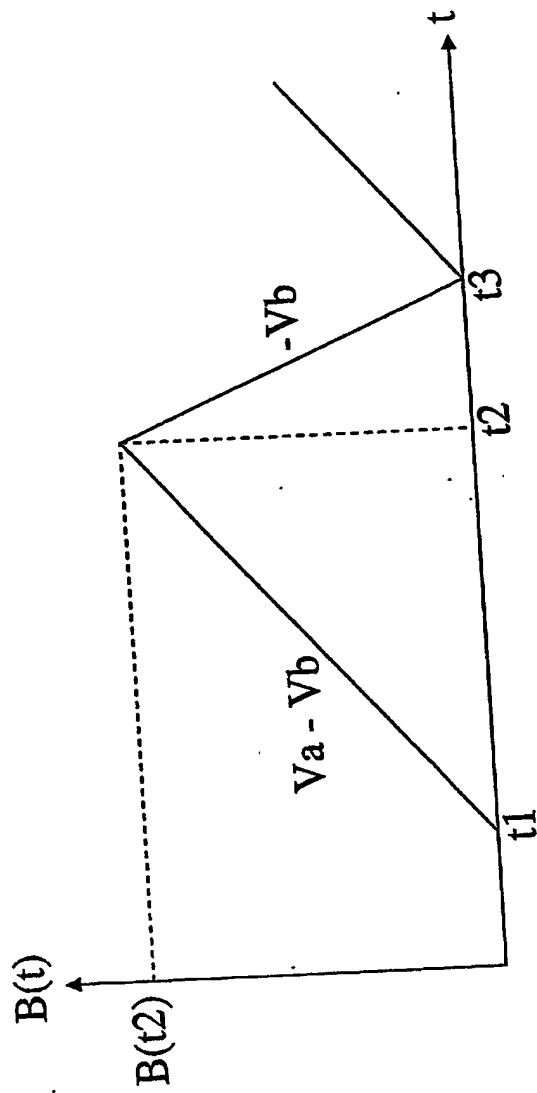
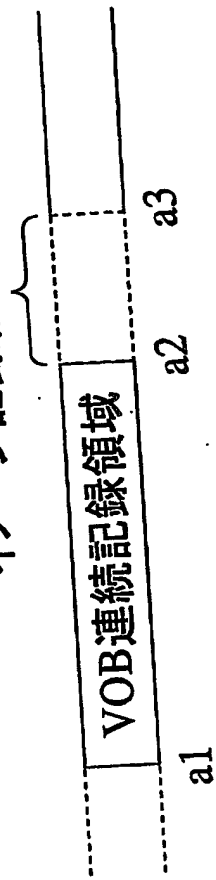


【図17】

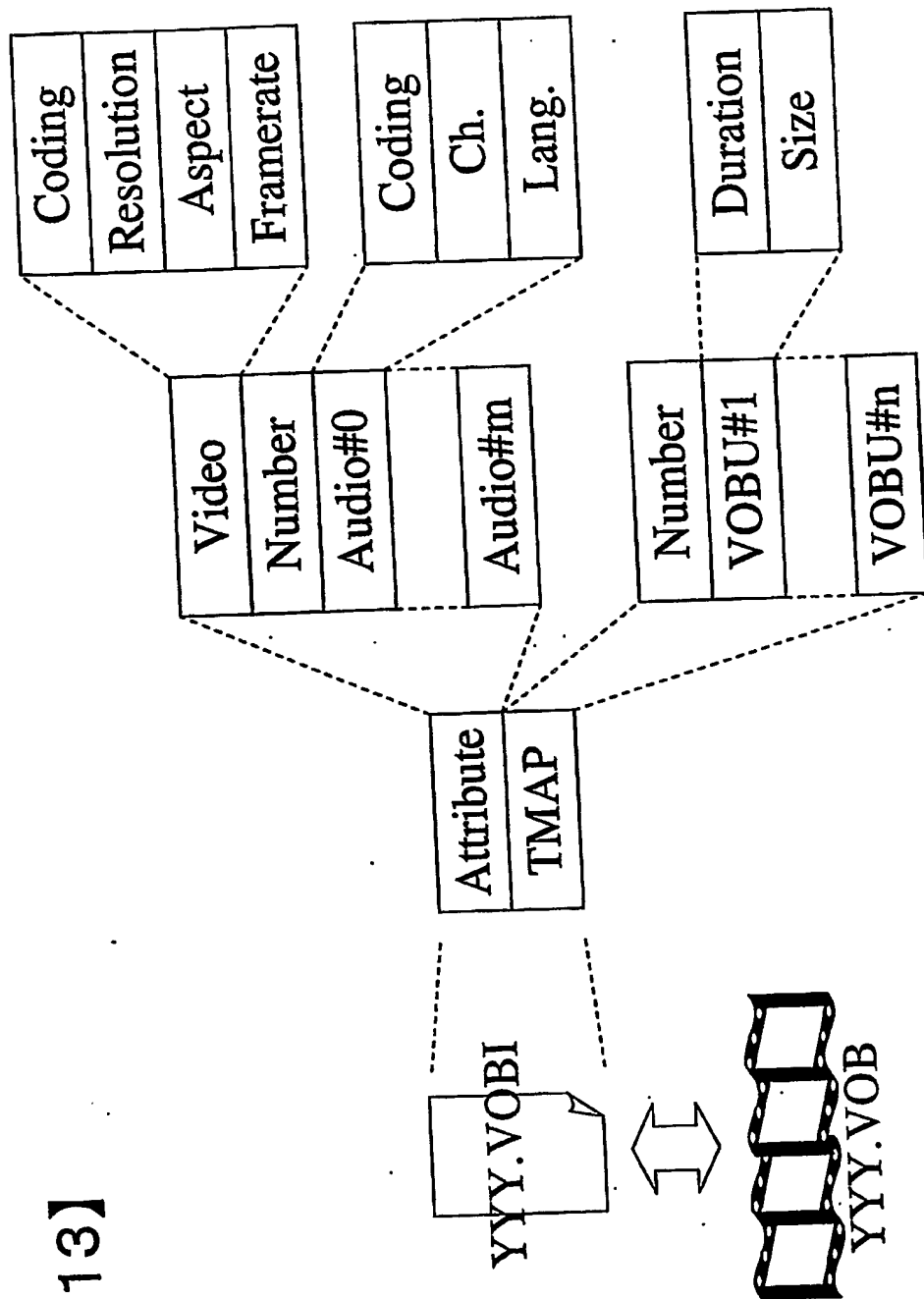


【図12】

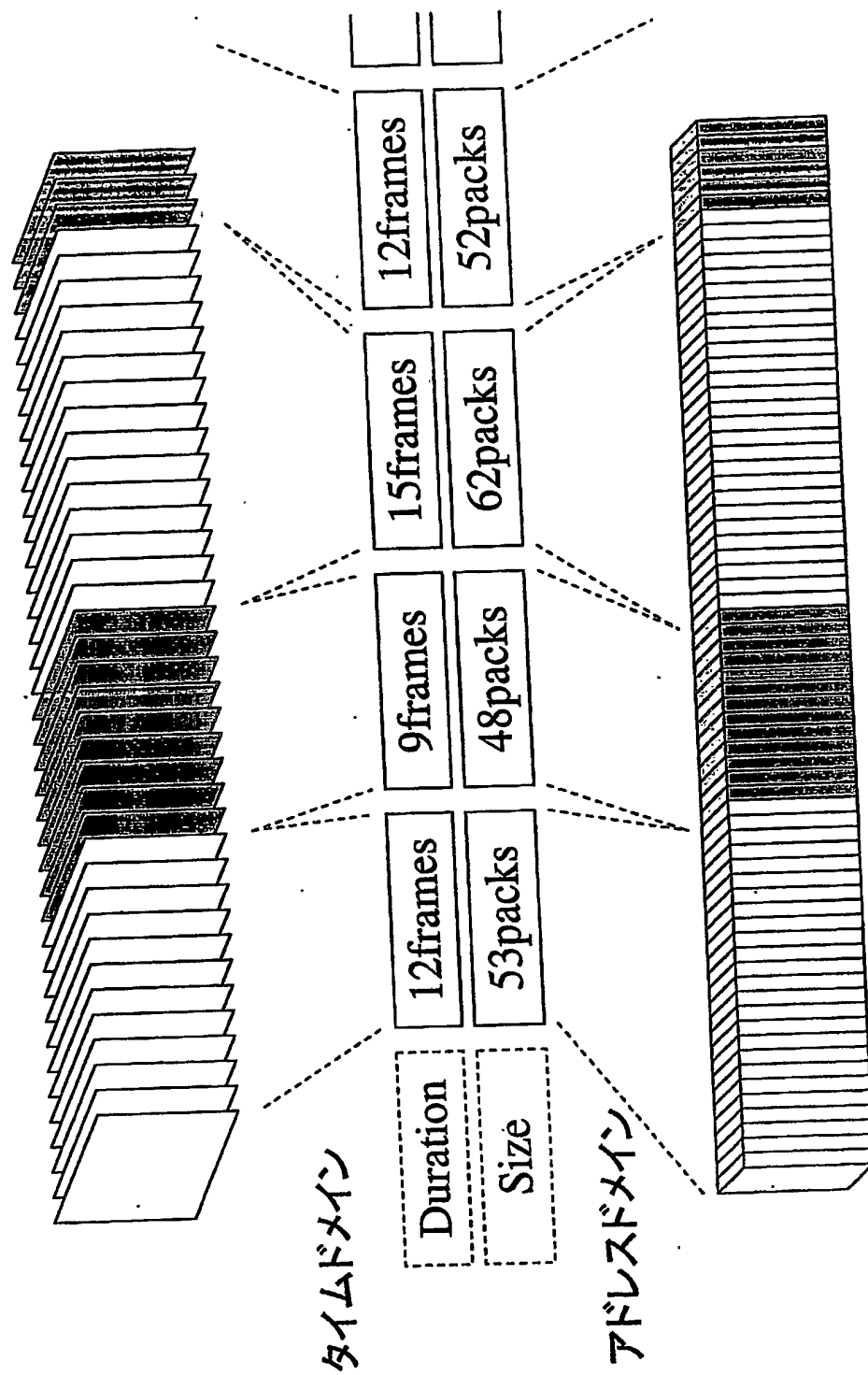
イメージ記録領域 or シーク



【図13】

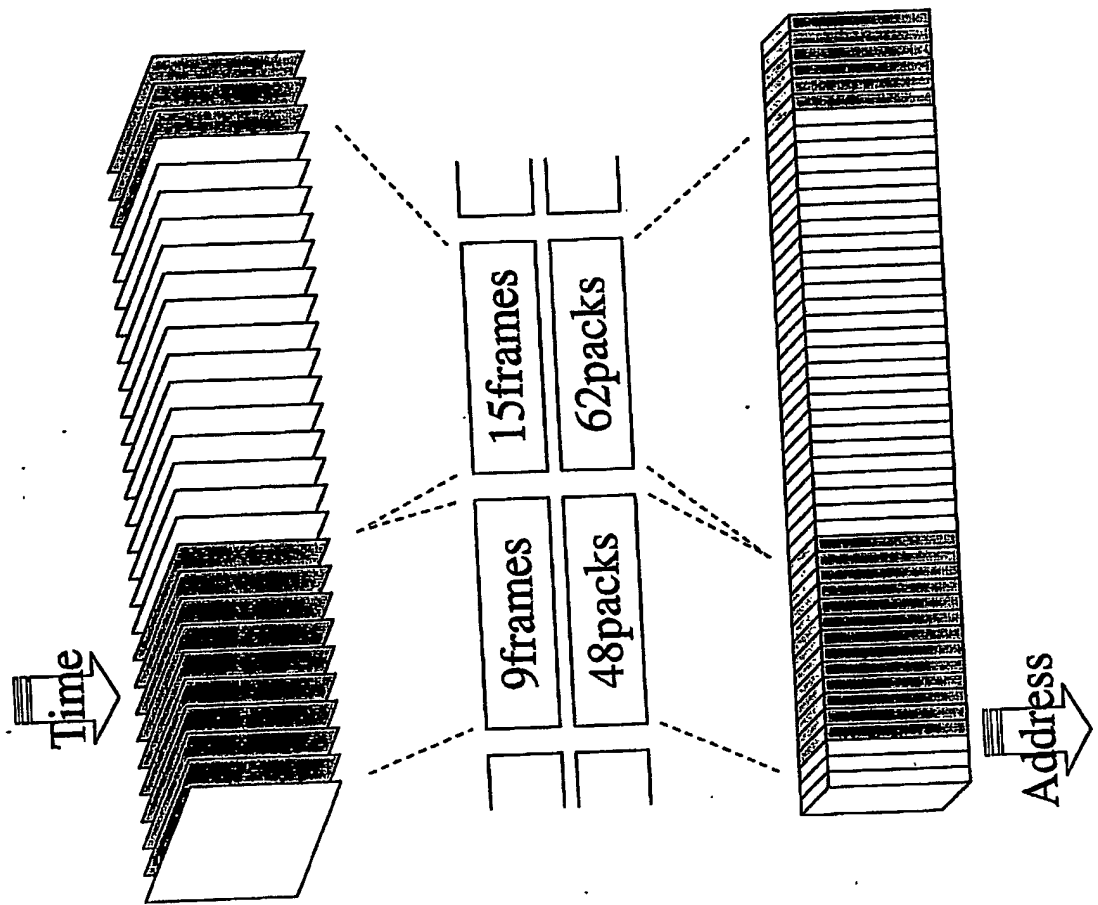


【図14】

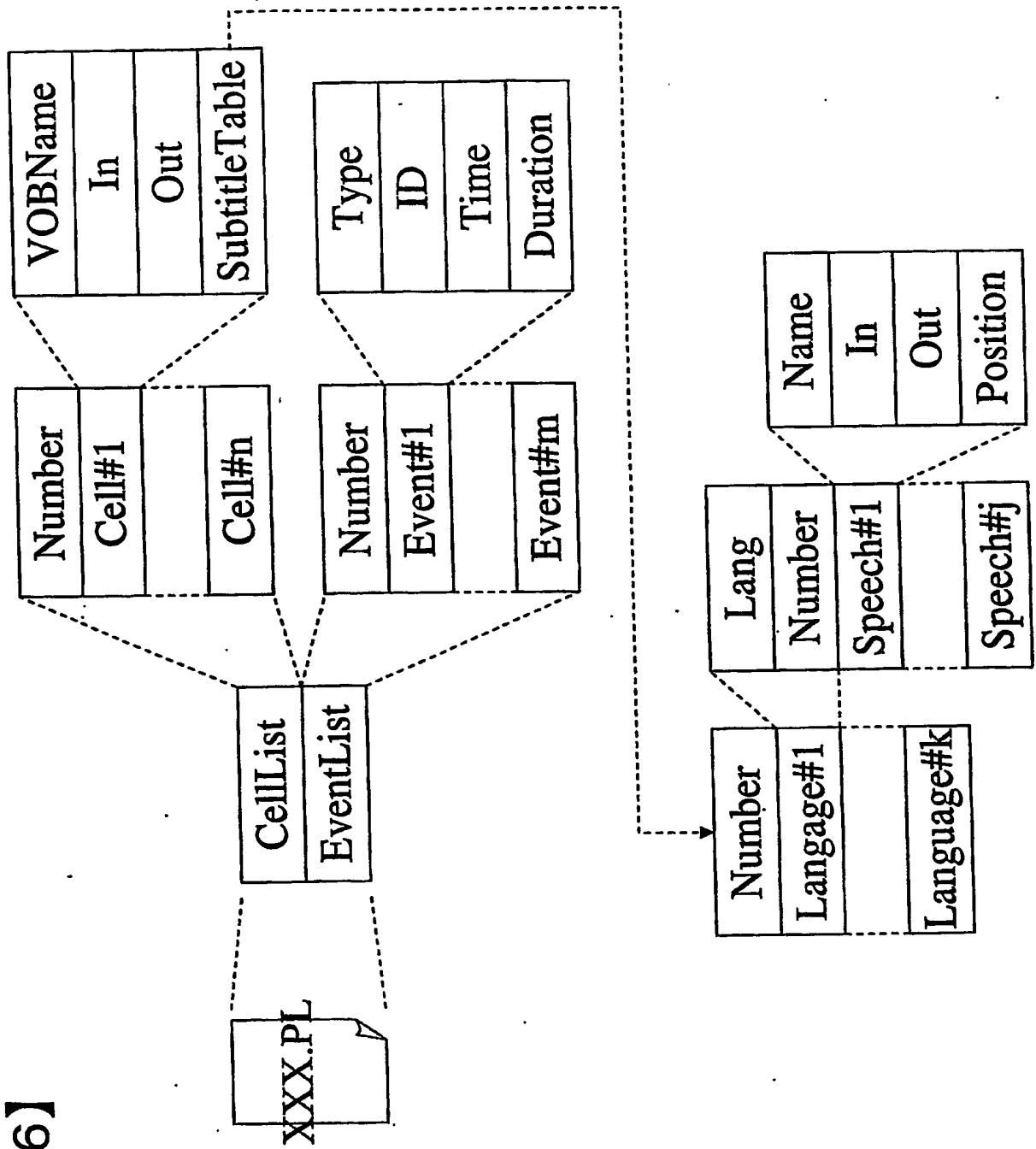




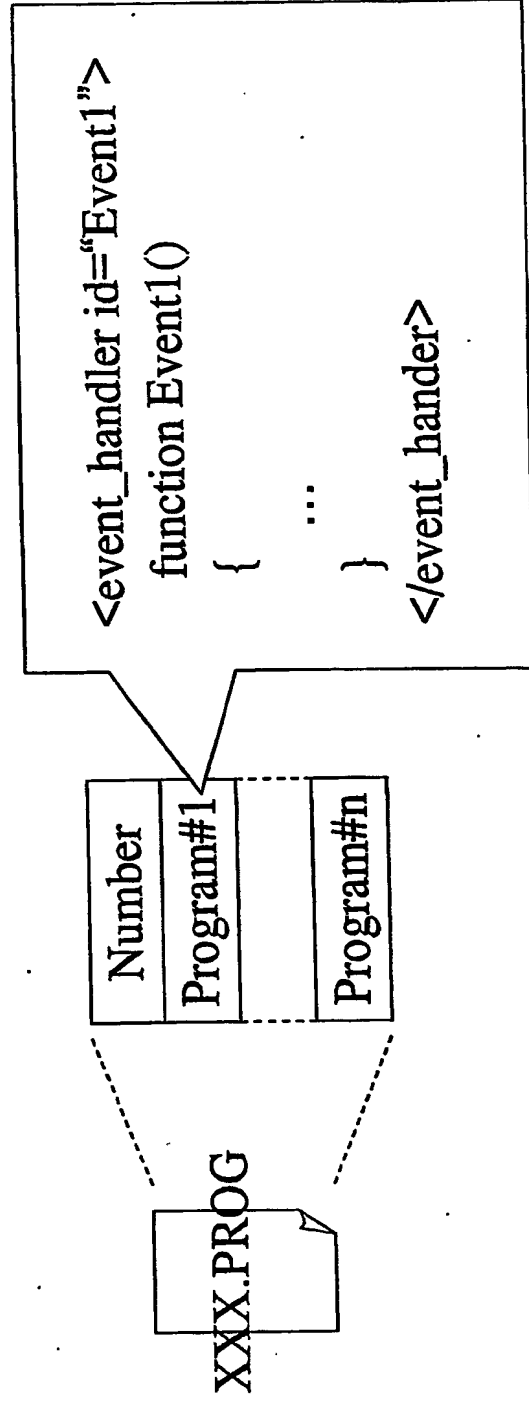
【图15】



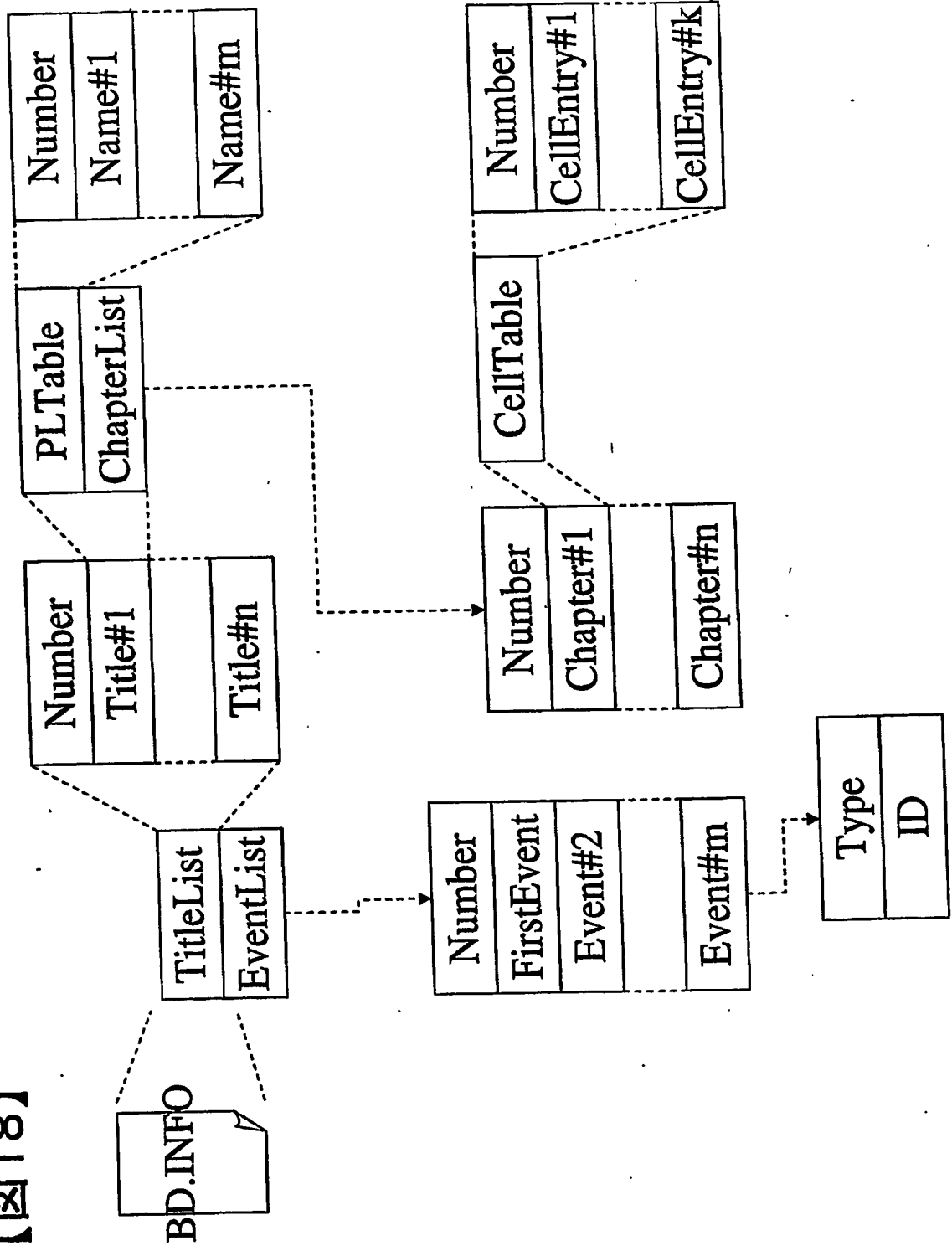
【図16】



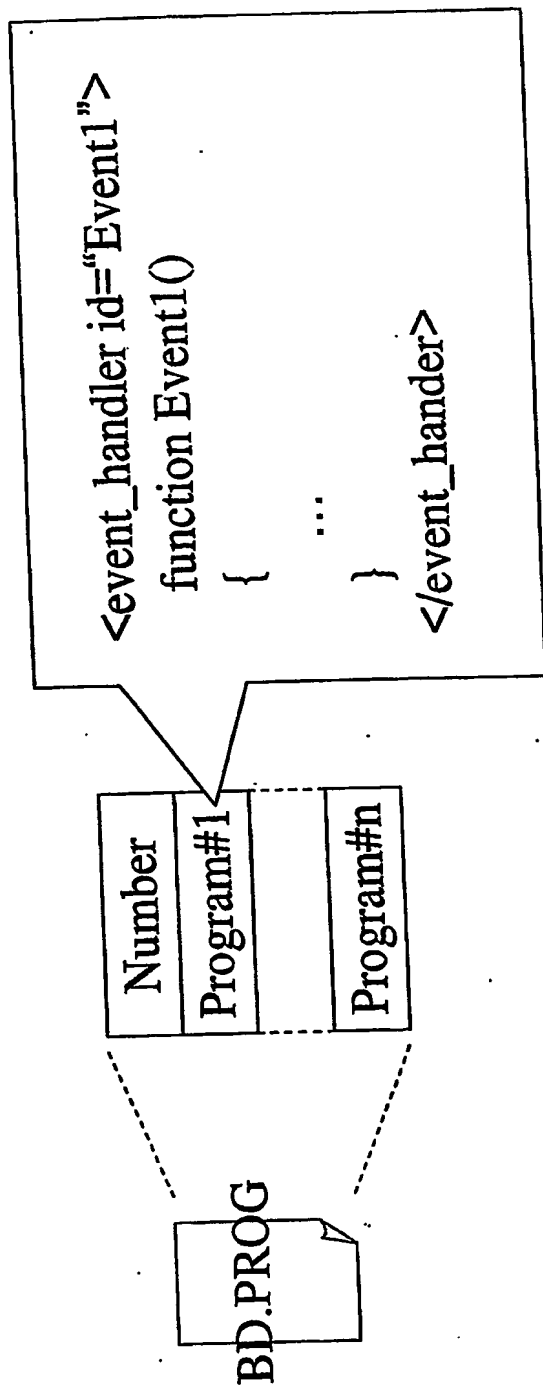
【图17】

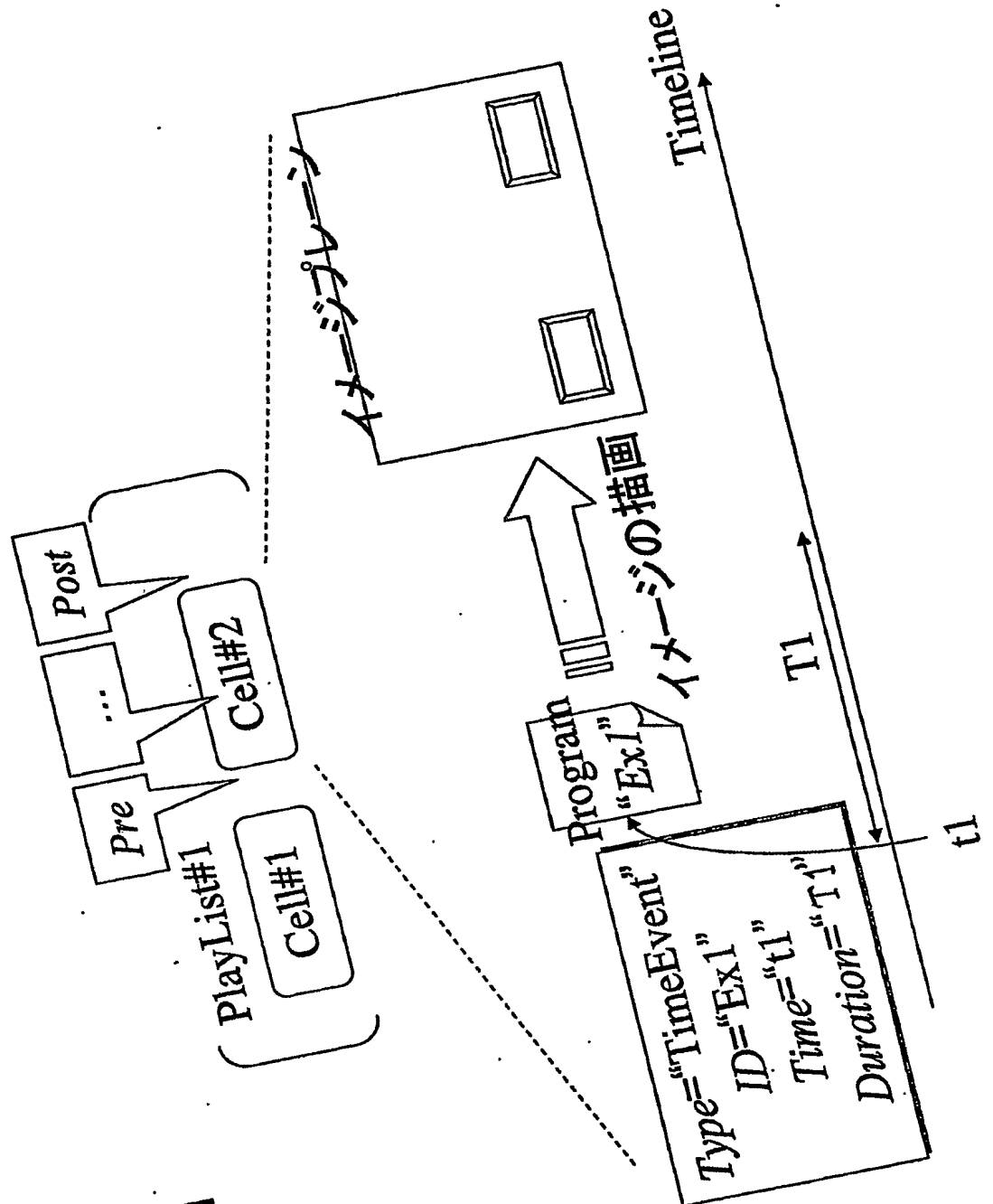


【図18】



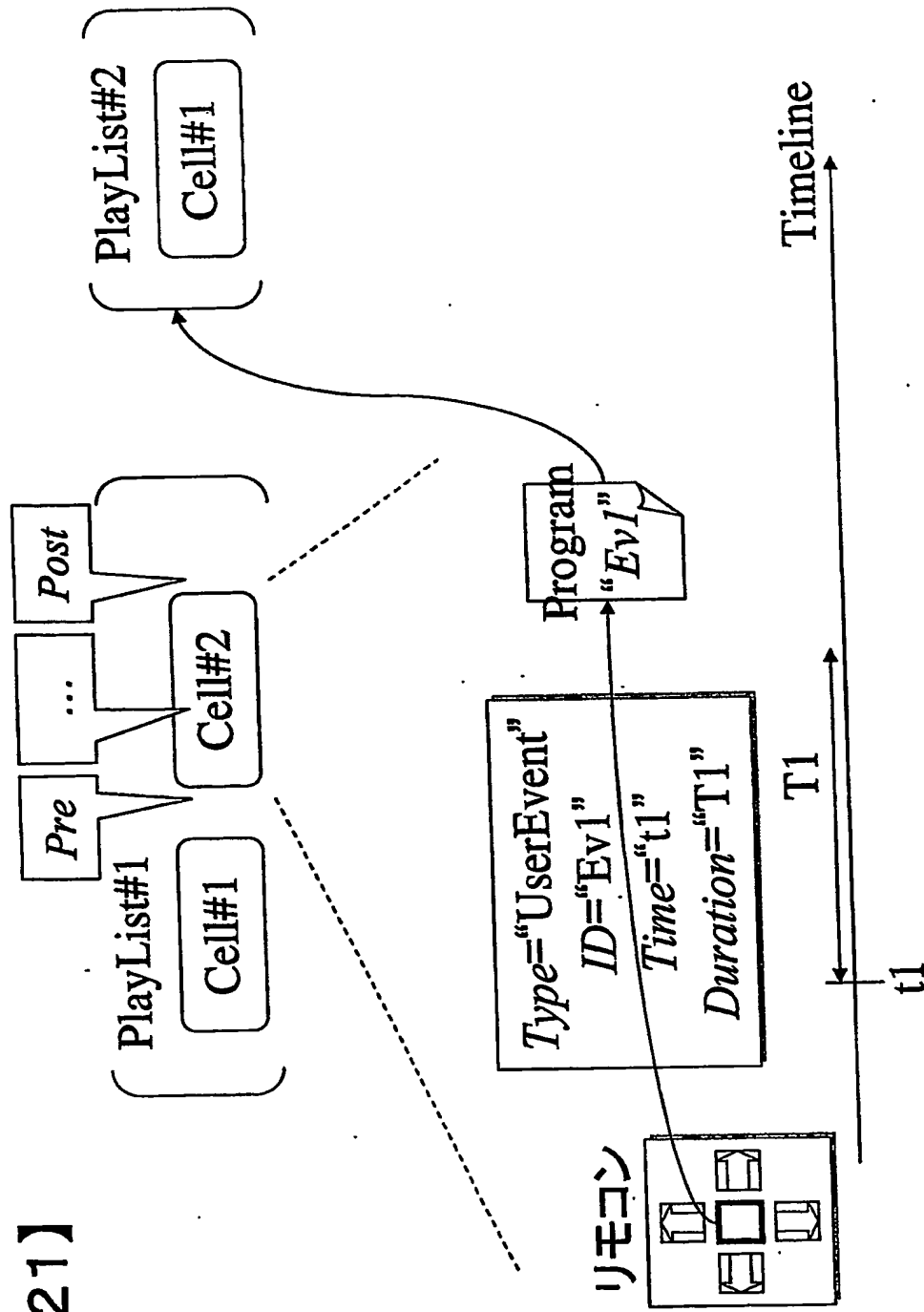
【图19】



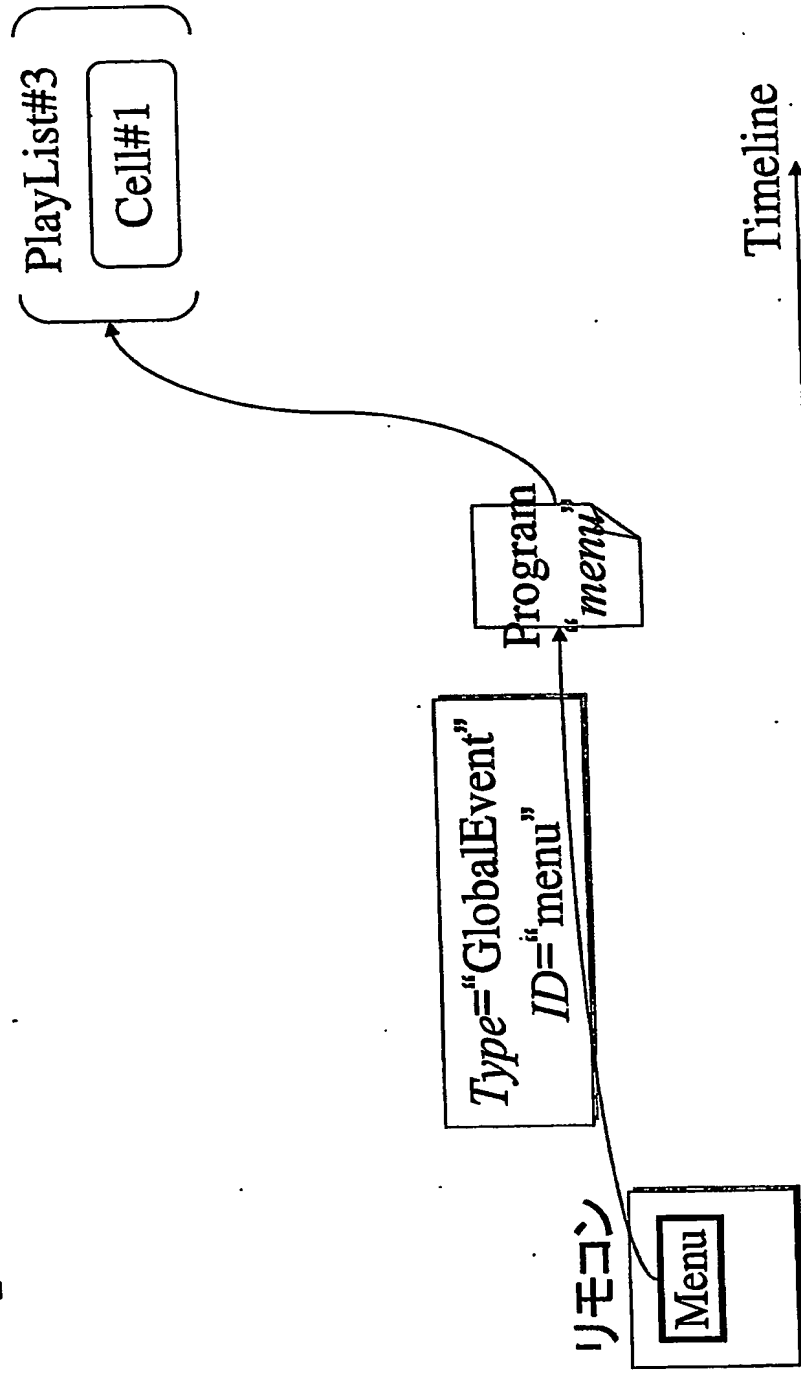


【図20】

【図21】



【図22】

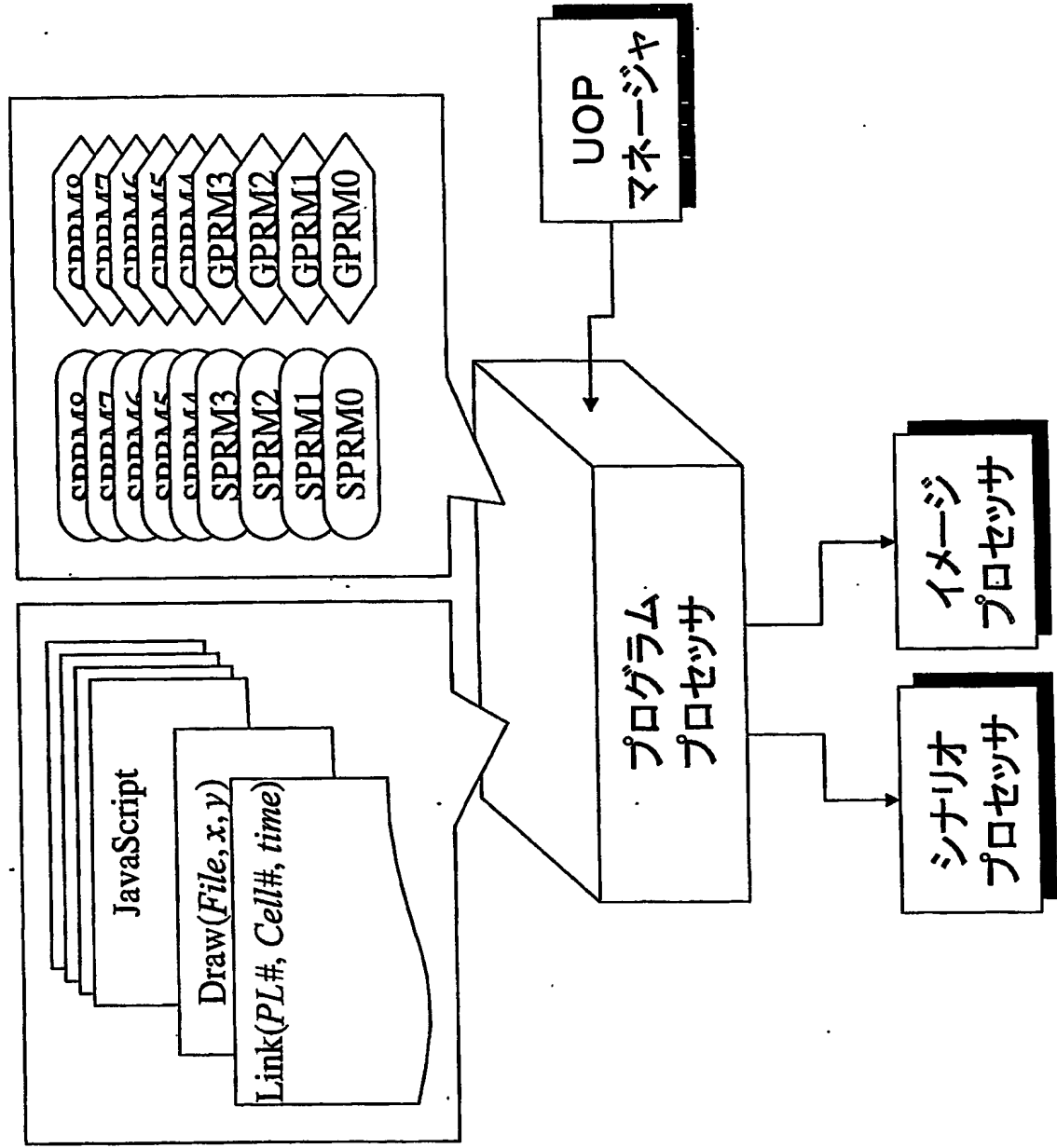




【図23】

プログラミング関数

プレーヤ変数

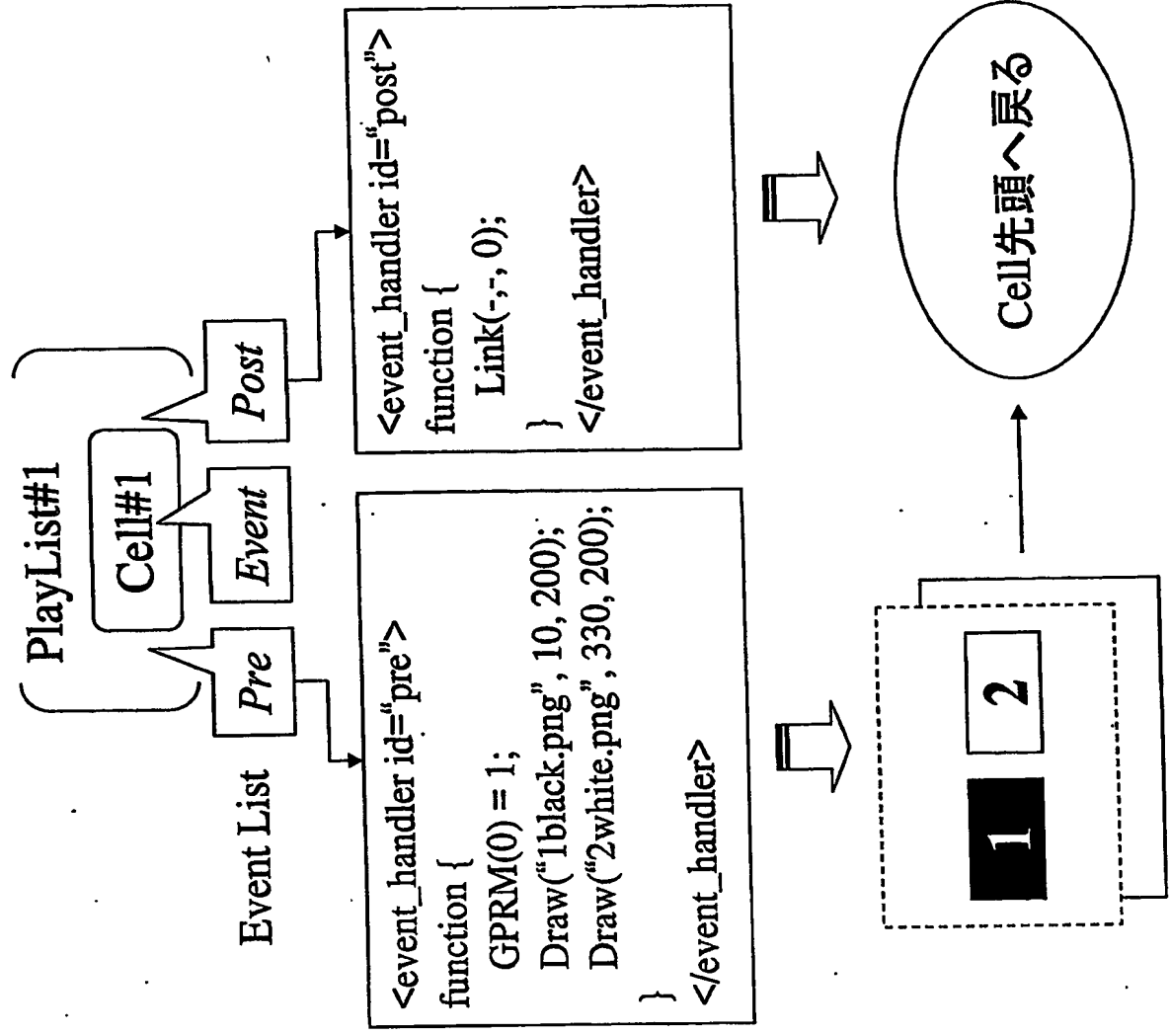


【図24】

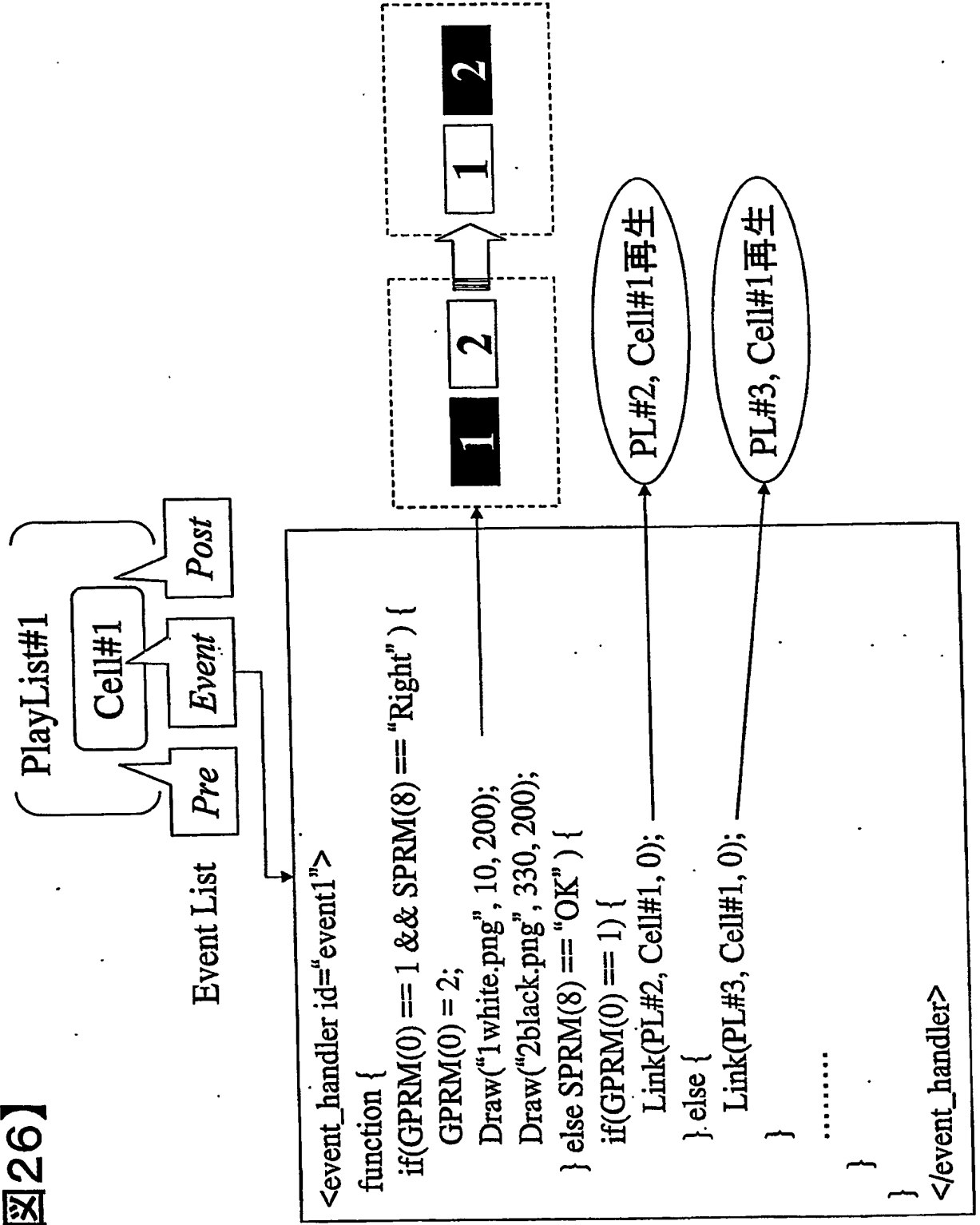
プレーヤー変数(システムパラメータ)

0	Language Code	11	Player audio mixing mode for Karaoke	22	reserved
1	Audio stream number	12	Country code for parental management	23	Player status
2	Subtitle stream number	13	Parental level	24	reserved
3	Angle number	14	Player configuration for Video	25	reserved
4	Title number	15	Player configuration for Audio	26	reserved
5	Chapter number	16	Language code for AST	27	reserved
6	Program number	17	Language code ext. for AST	28	reserved
7	Cell number	18	Language code for STST	29	reserved
8	Key name	19	Language coded ext. for STST	30	reserved
9	Navigation timer	20	Player region code	31	reserved
10	Current playback time	21	reserved	32	reserved

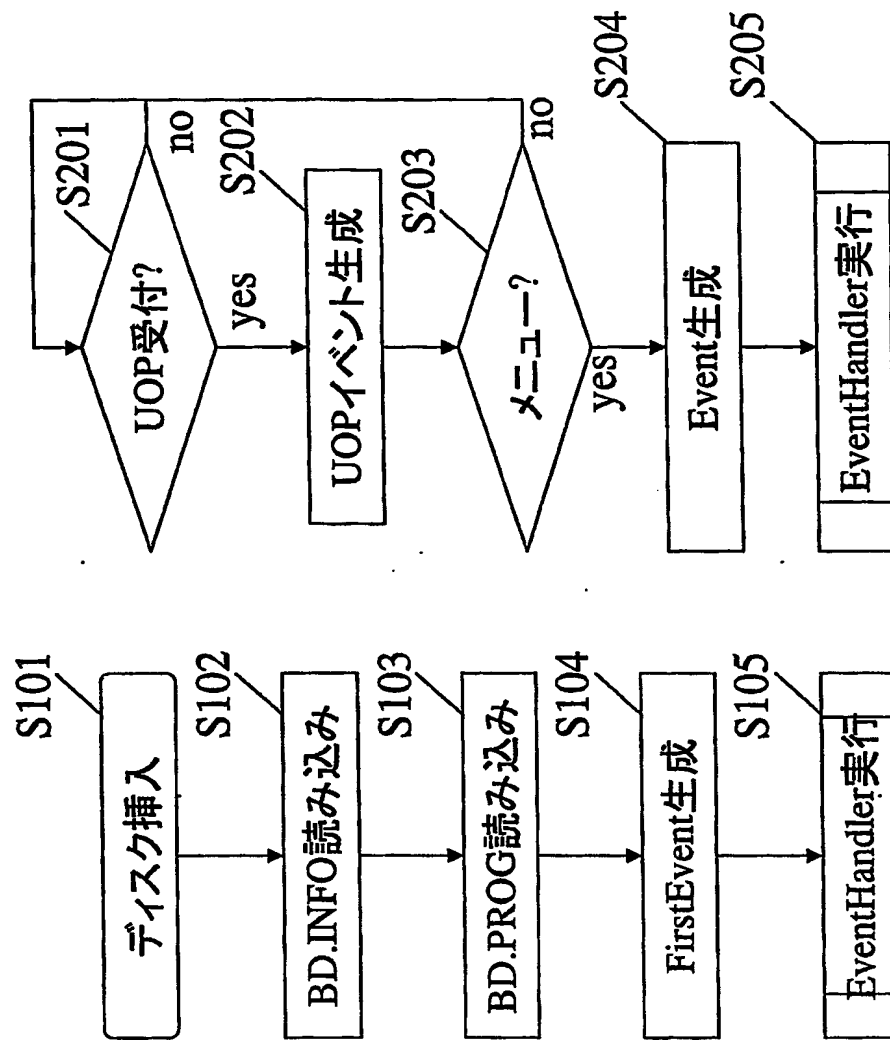
【図25】



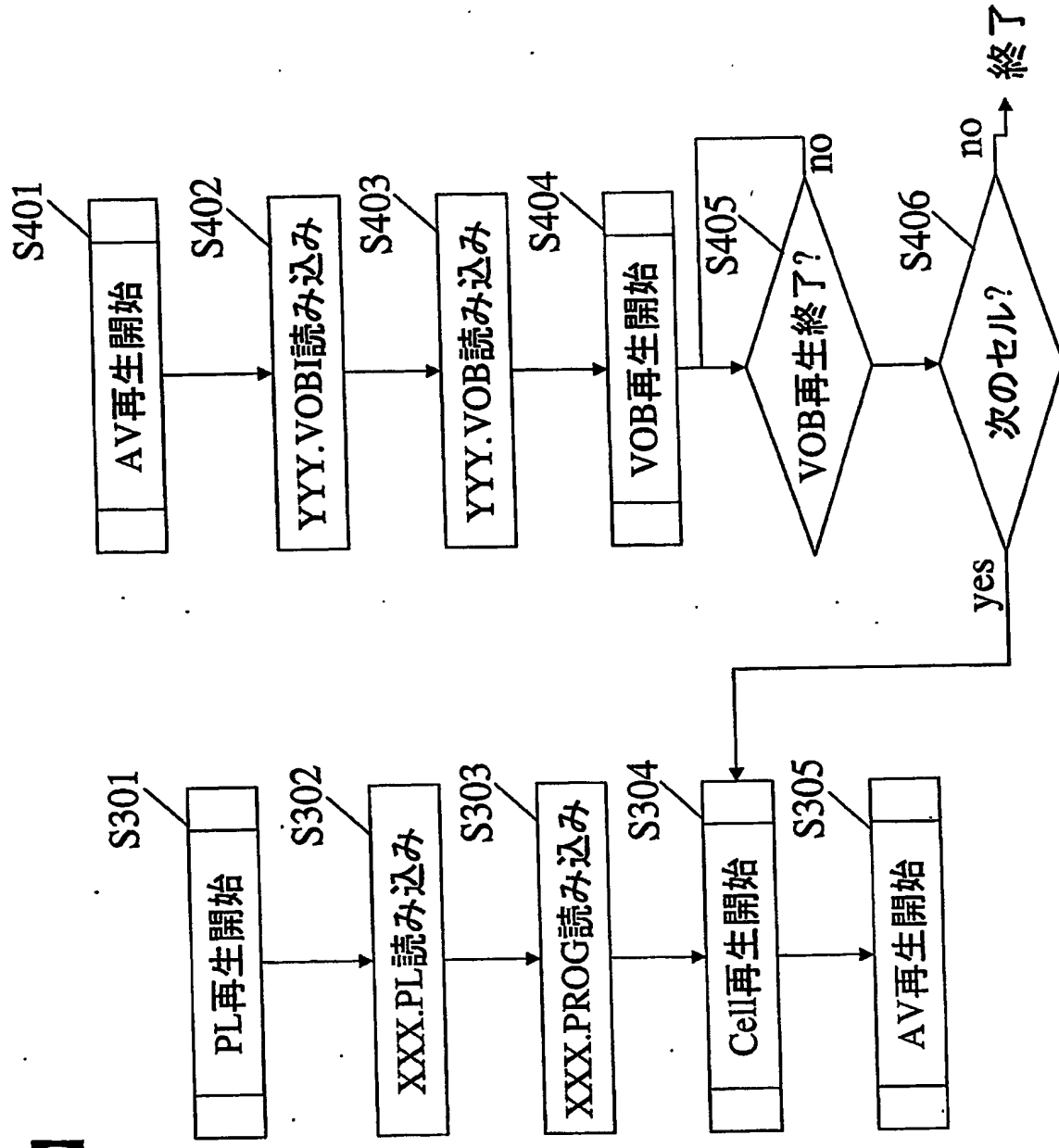
【図26】



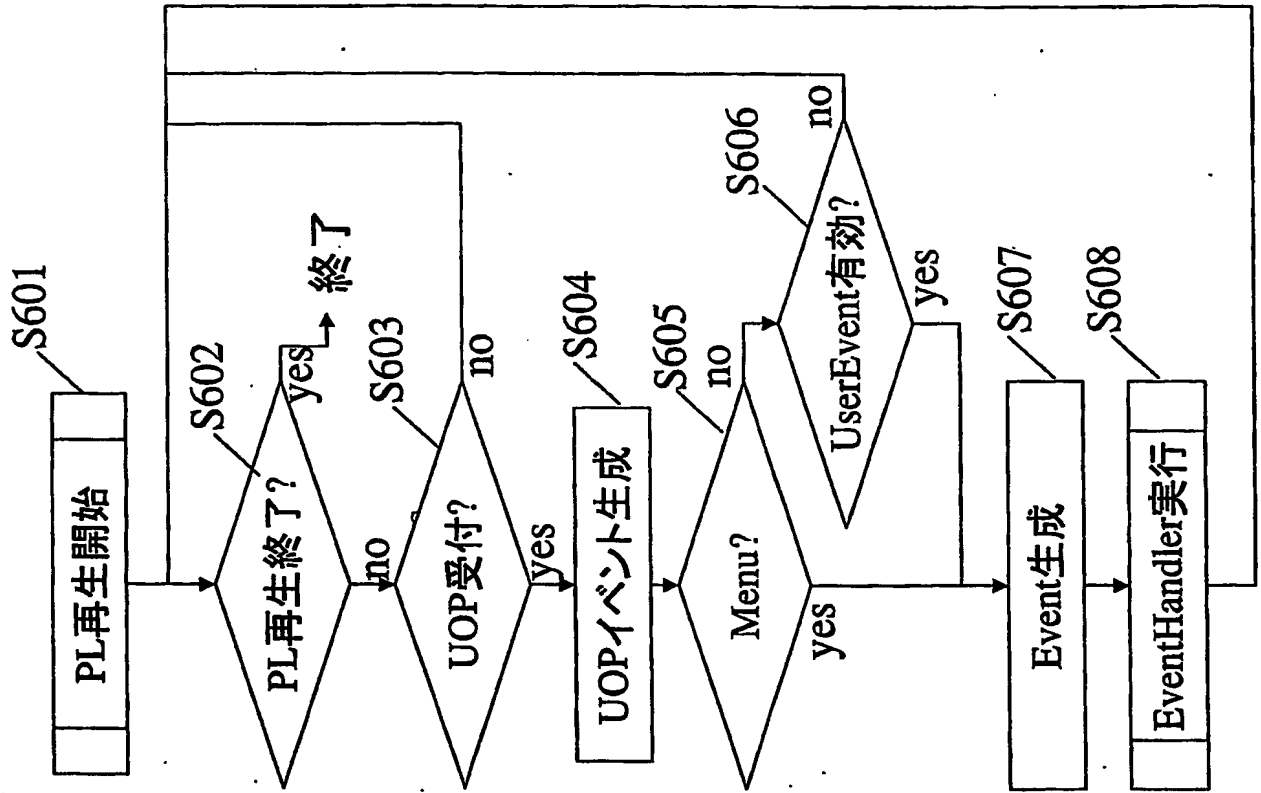
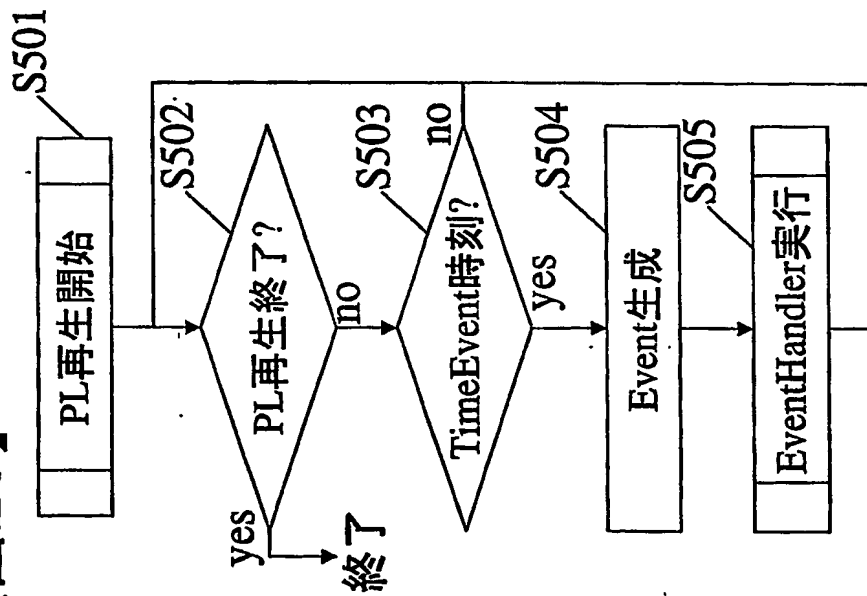
【図27】



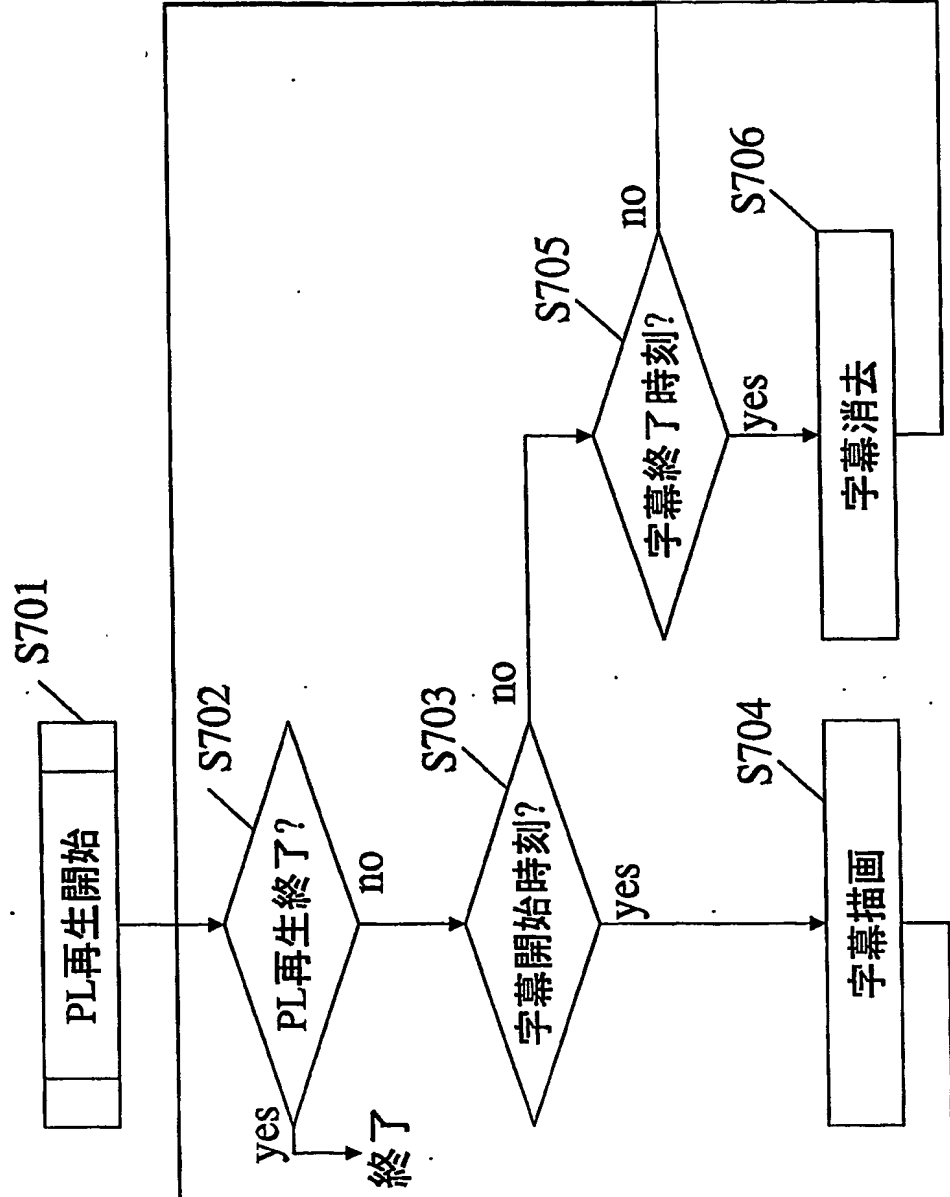
【図28】



【図29】

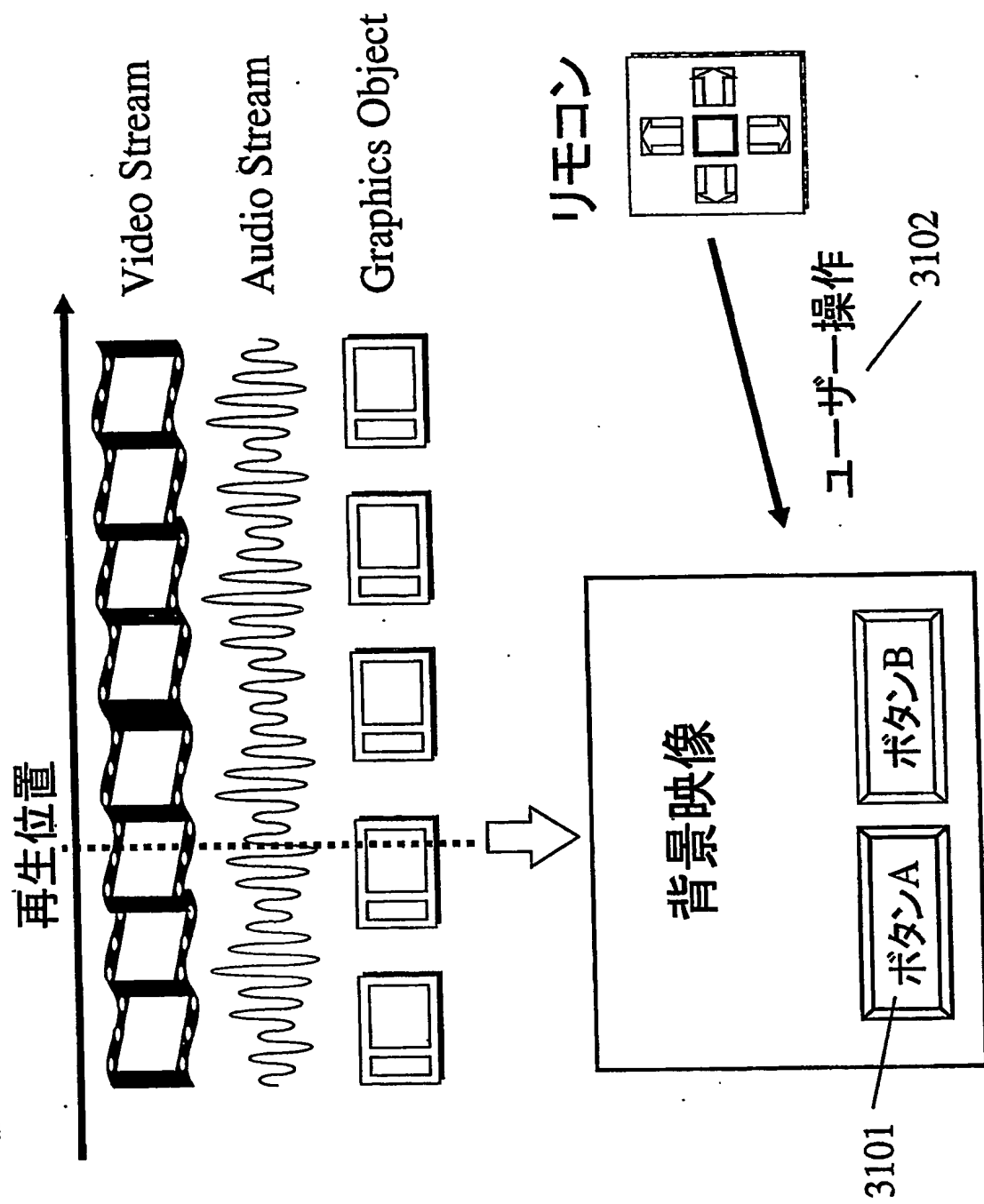


【図30】

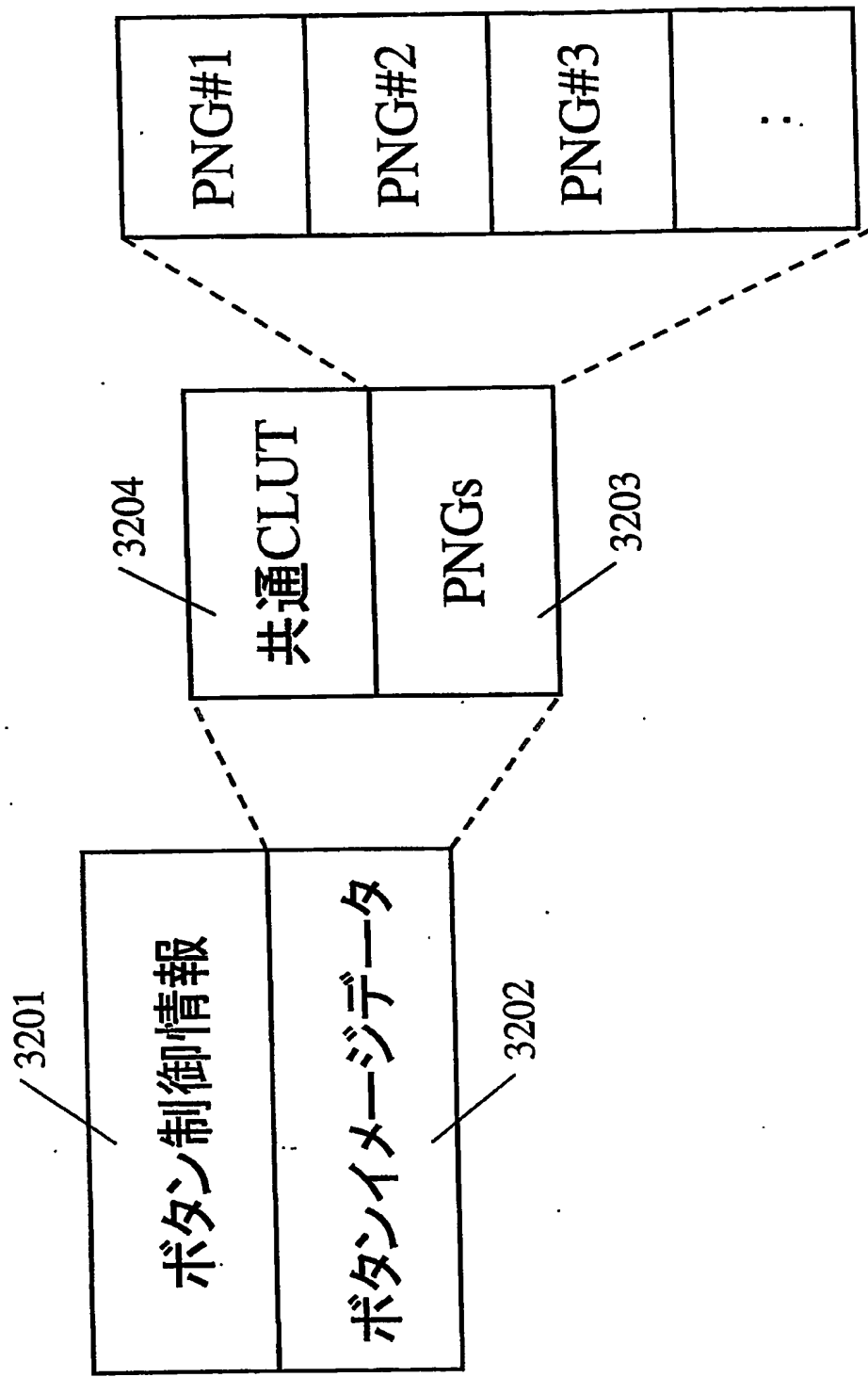




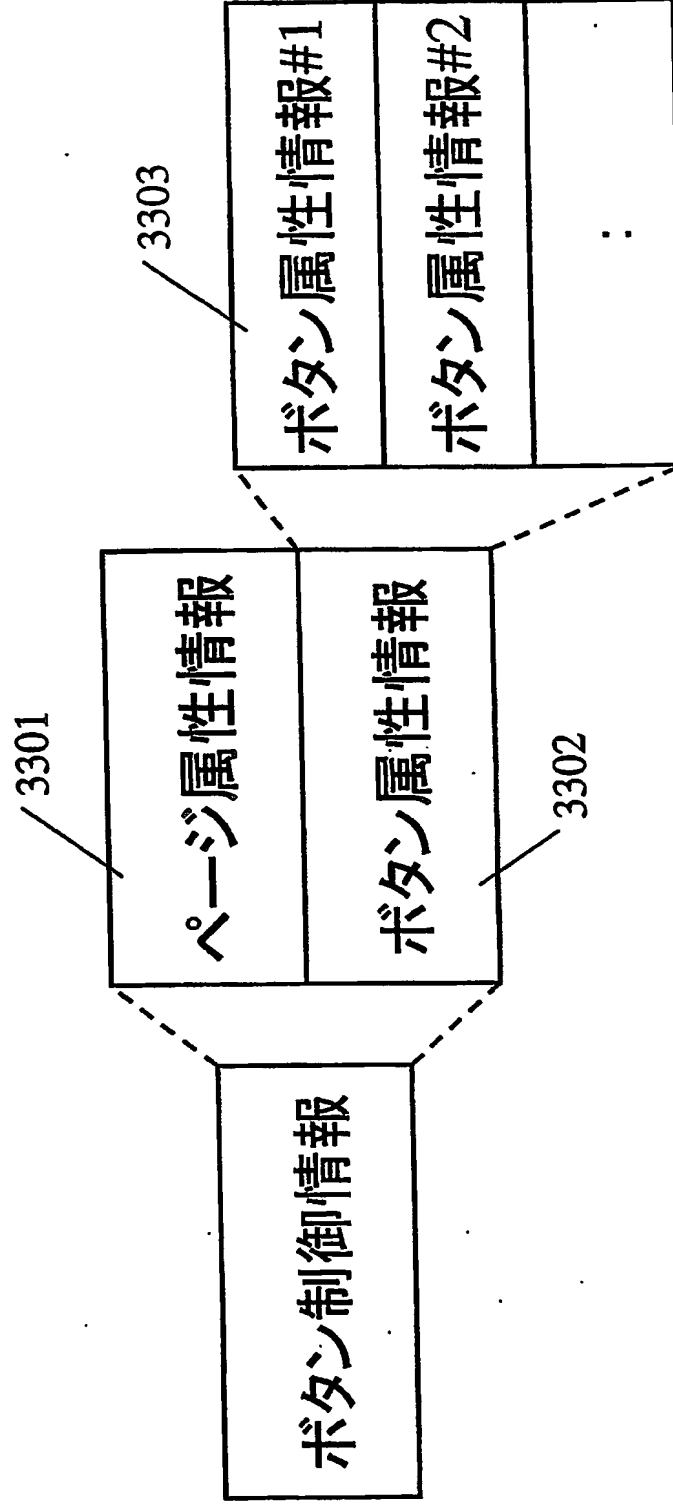
【図31】



【図32】



【図33】



【図34】

# ページ属性情報

3401	status_of_Highlight
3402	start_PTS_of_Highlight
3403	end_PTS_of_Highlight
3404	end_PTS_of_button_selection
3405	num_of_button_groups
3406	button_display_types
3408	button_offset_number
3409	num_of_buttons_in_group
3410	num_of_numerical_select_button
3411	forcedly_selected_button
3412	forcedly_activated_button

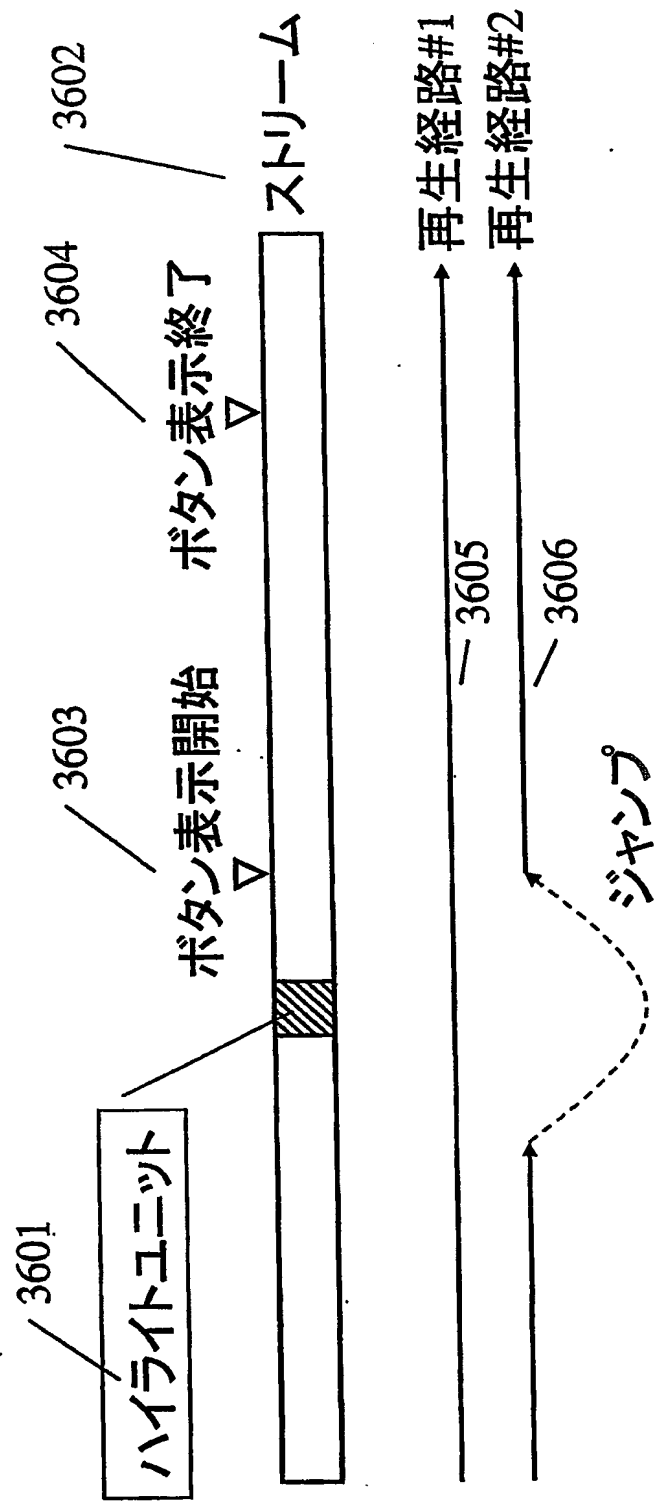
3407	button_display_type #1
	button_display_type #2
	..

【図35】

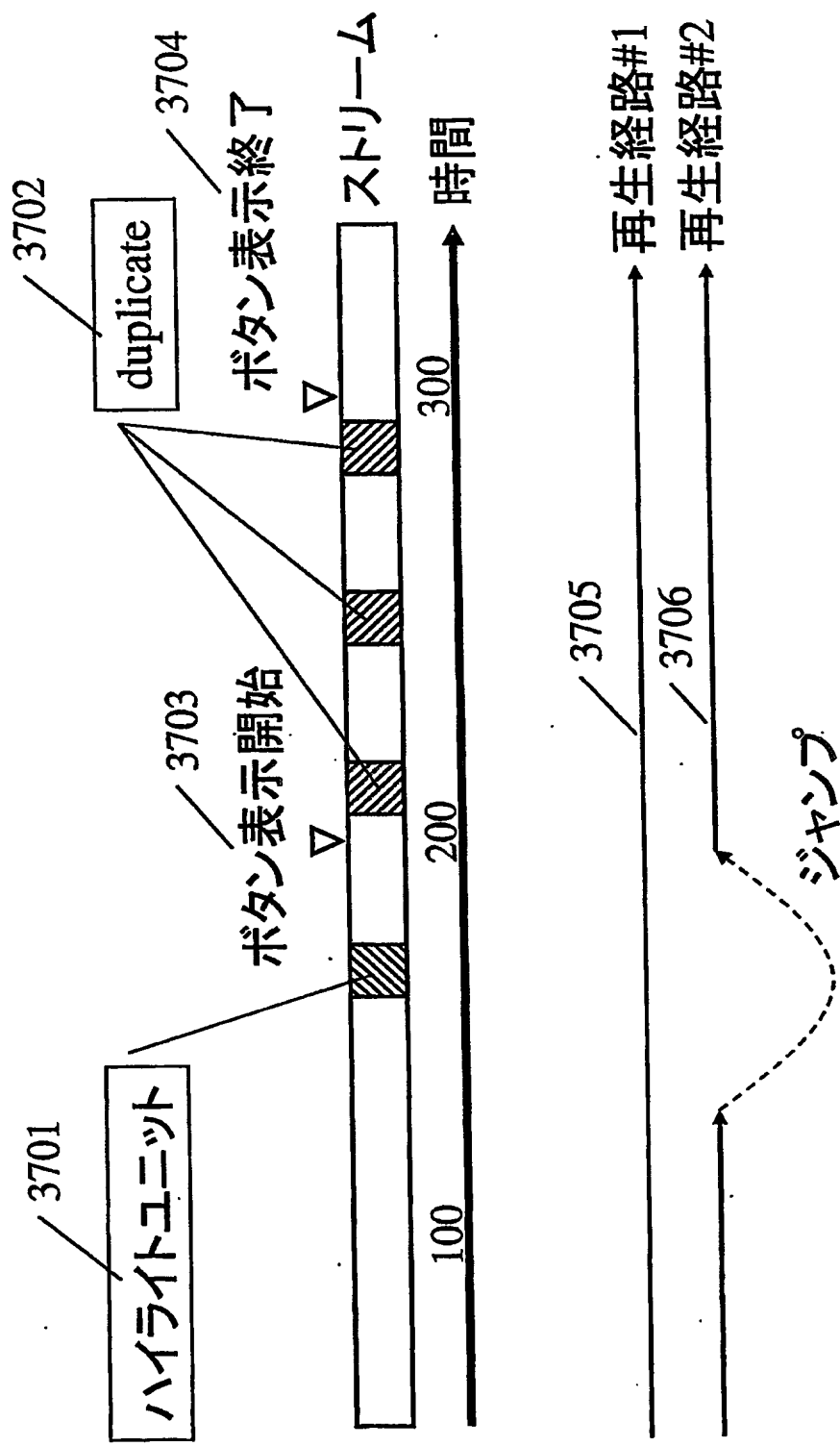
ボタン属性情報

3501	normal_button_filename
3502	selection_button_filename
3503	action_button_filename
3504	start_x_coordinate
3505	end_x_coordinate
3506	start_y_coordinate
3507	end_y_coordinate
3508	auto_action_mode
3509	upper_button_id
3510	lower_button_id
3511	left_button_id
3512	right_button_id
3513	button_command

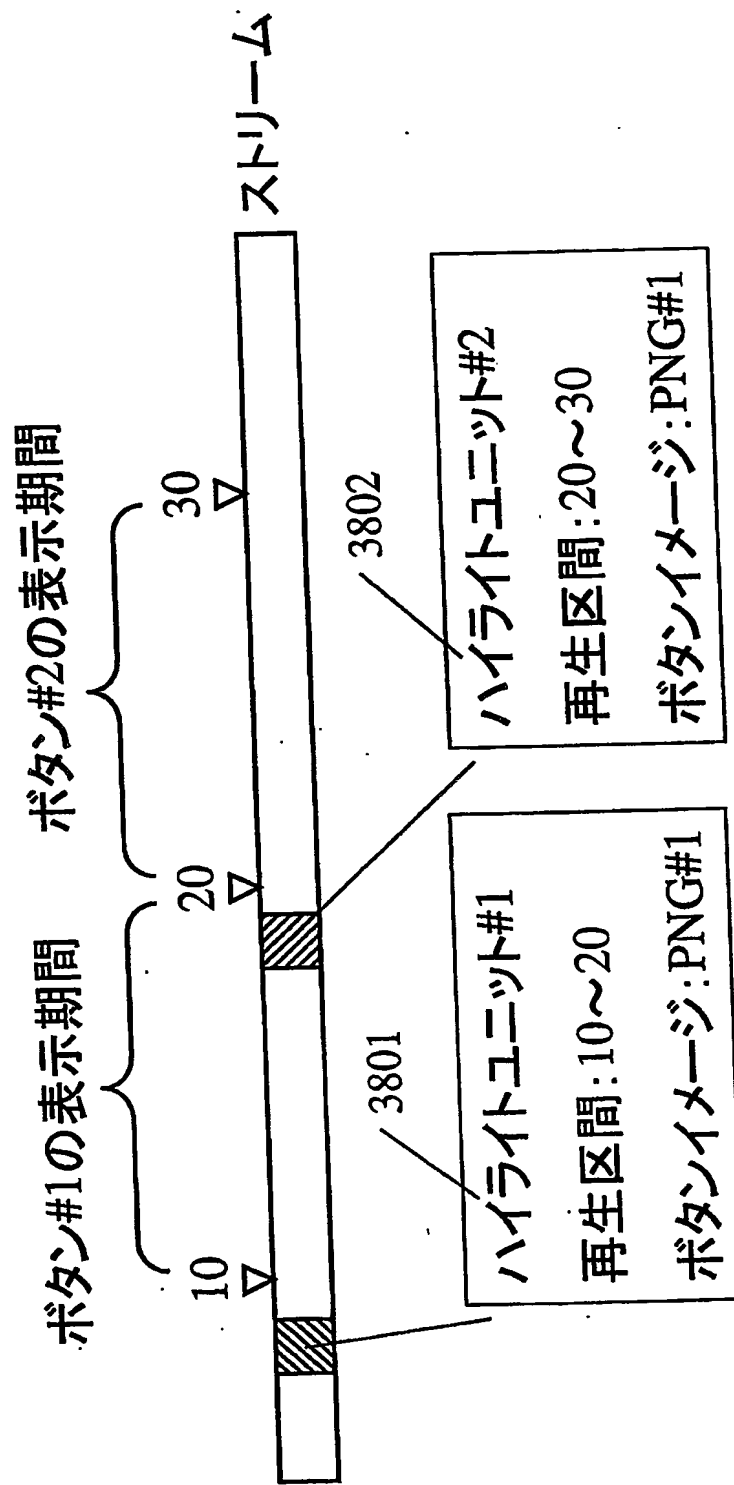
【図36】



【図37】

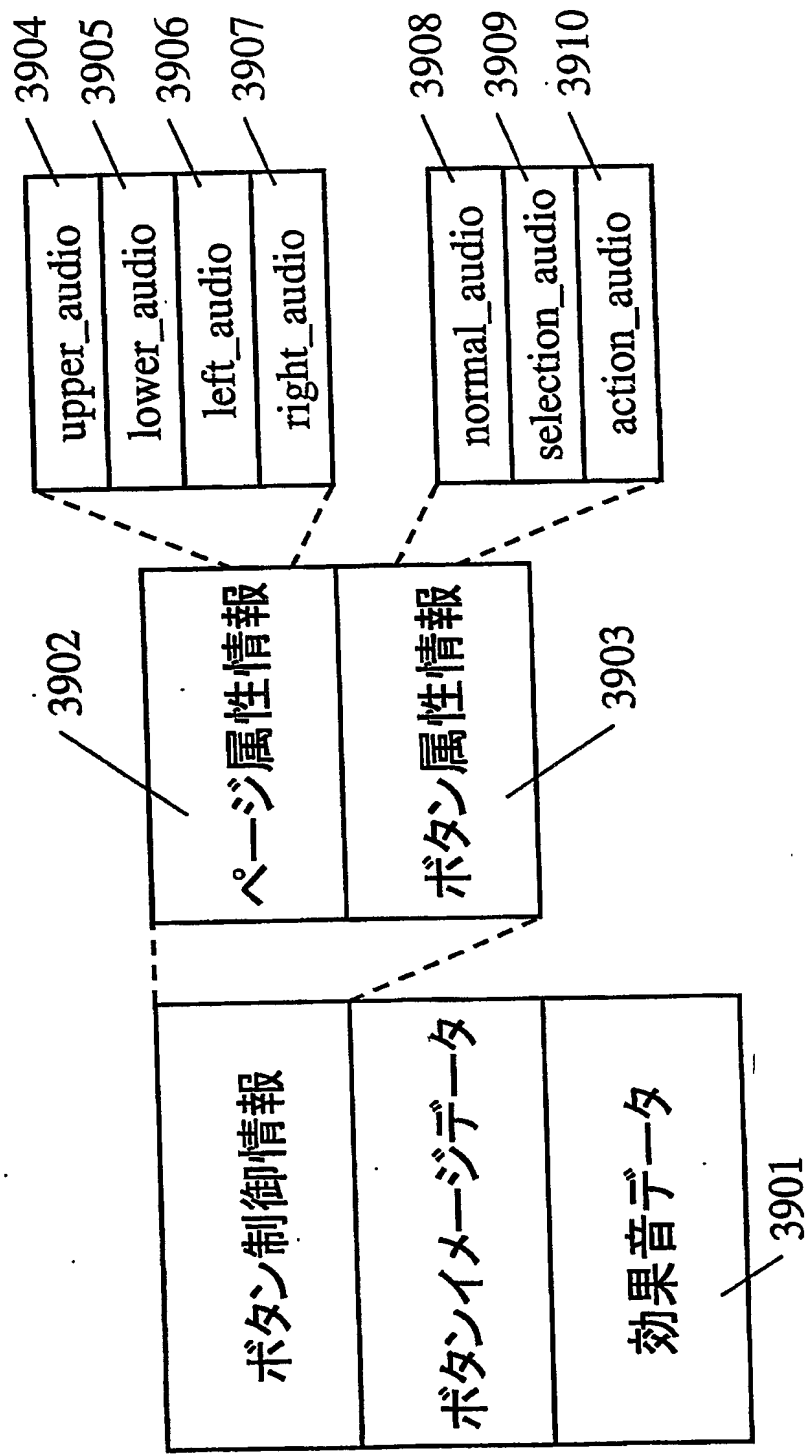


【図38】

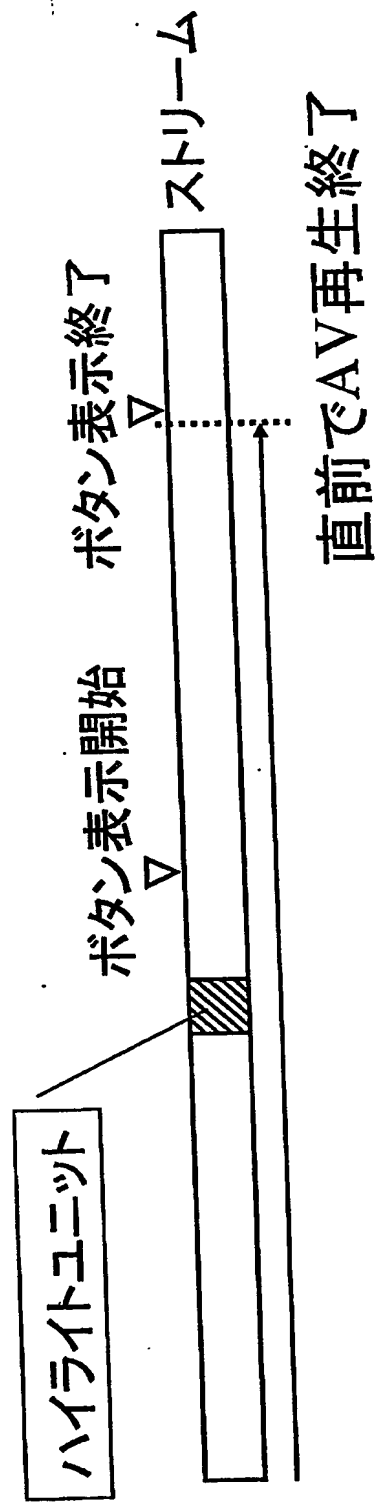




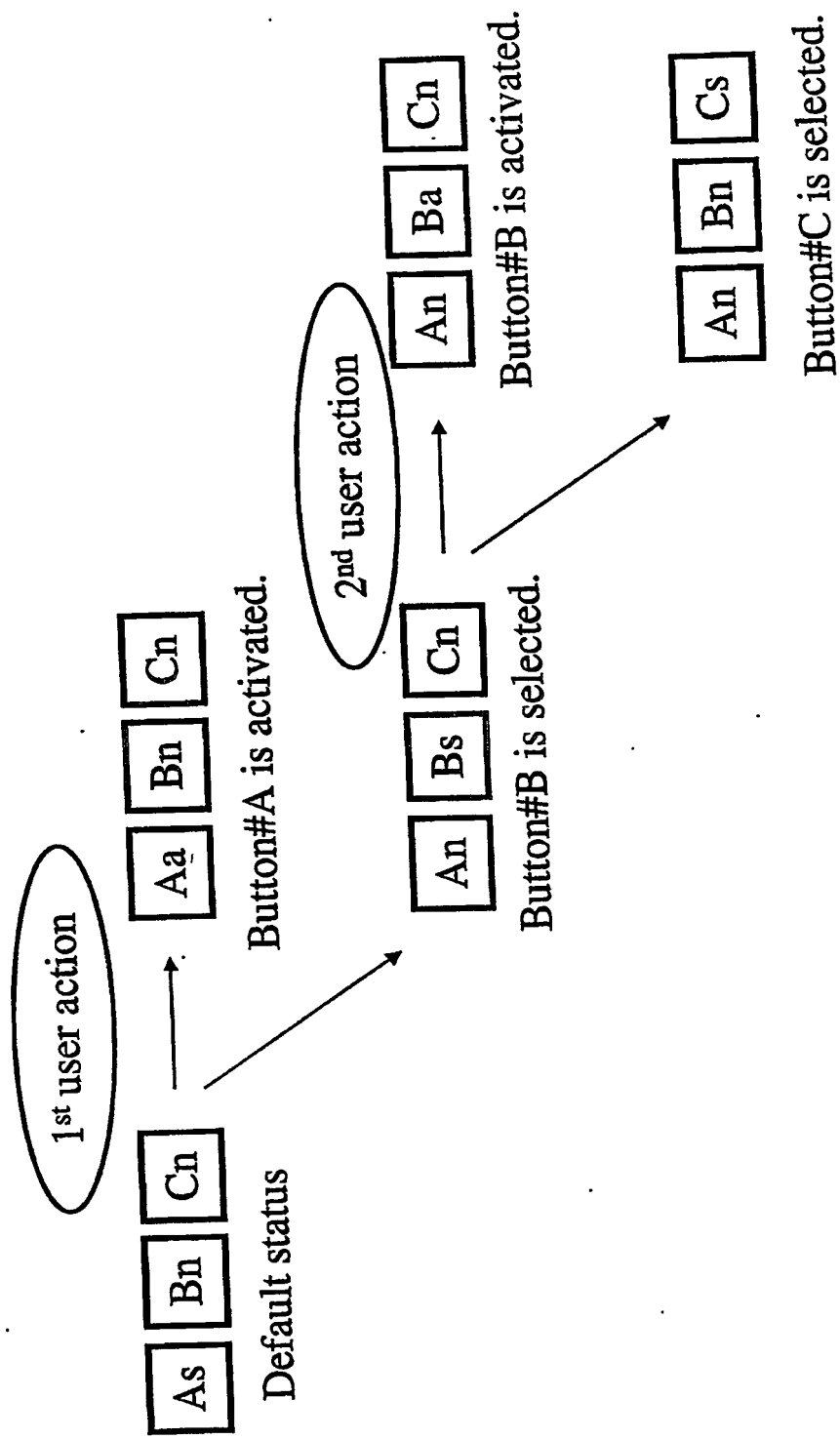
【図39】



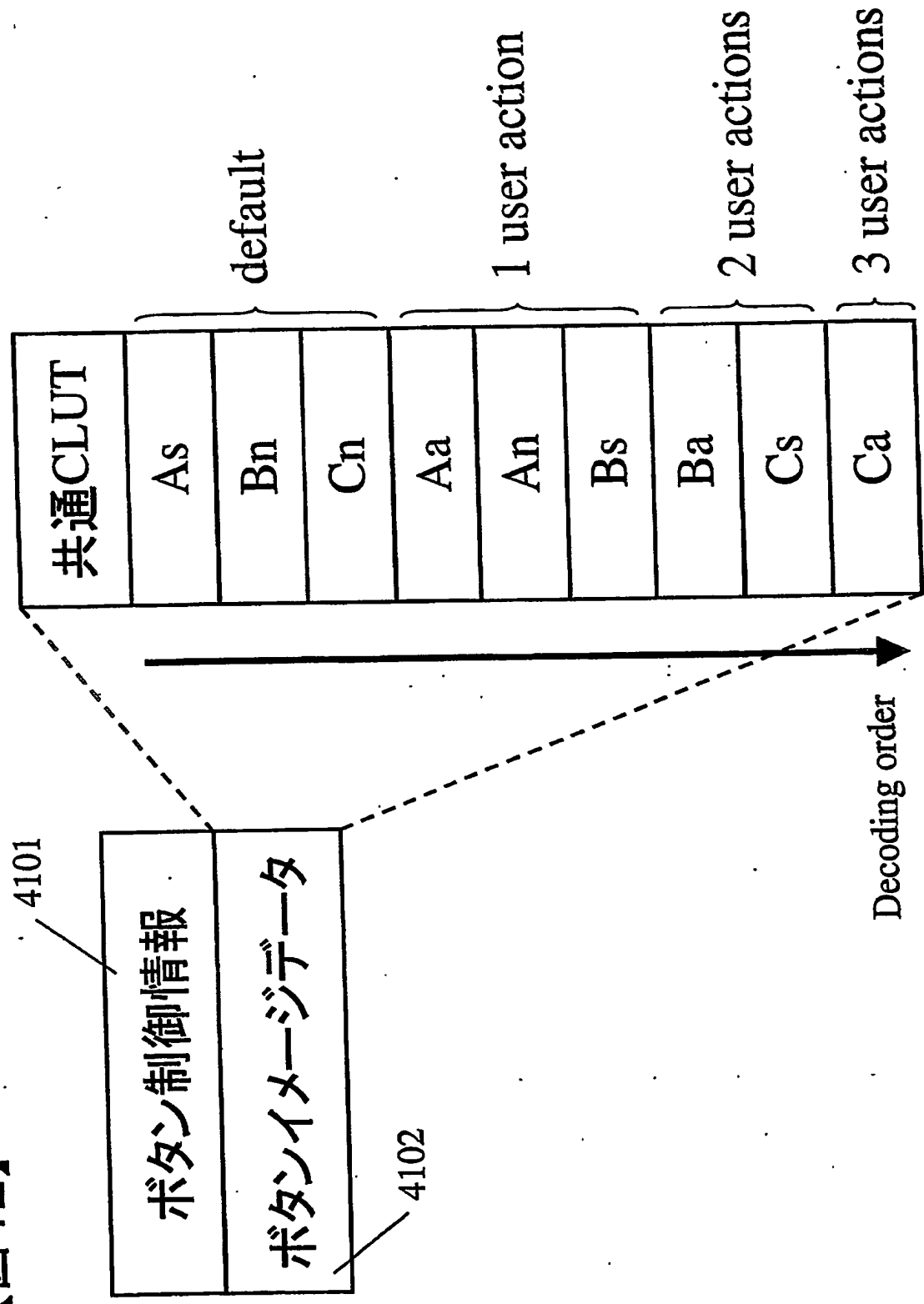
【図40】



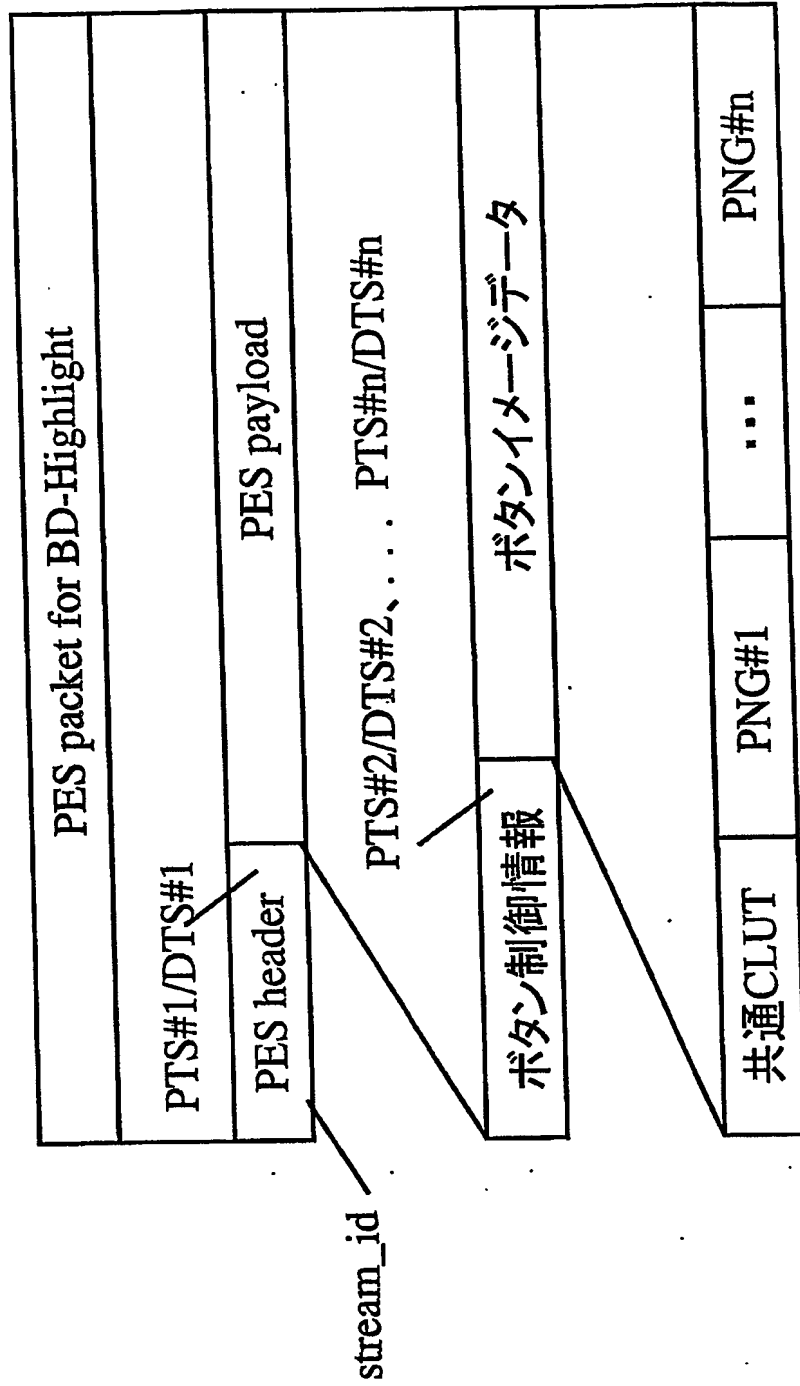
【图41】



【図42】



【図43】



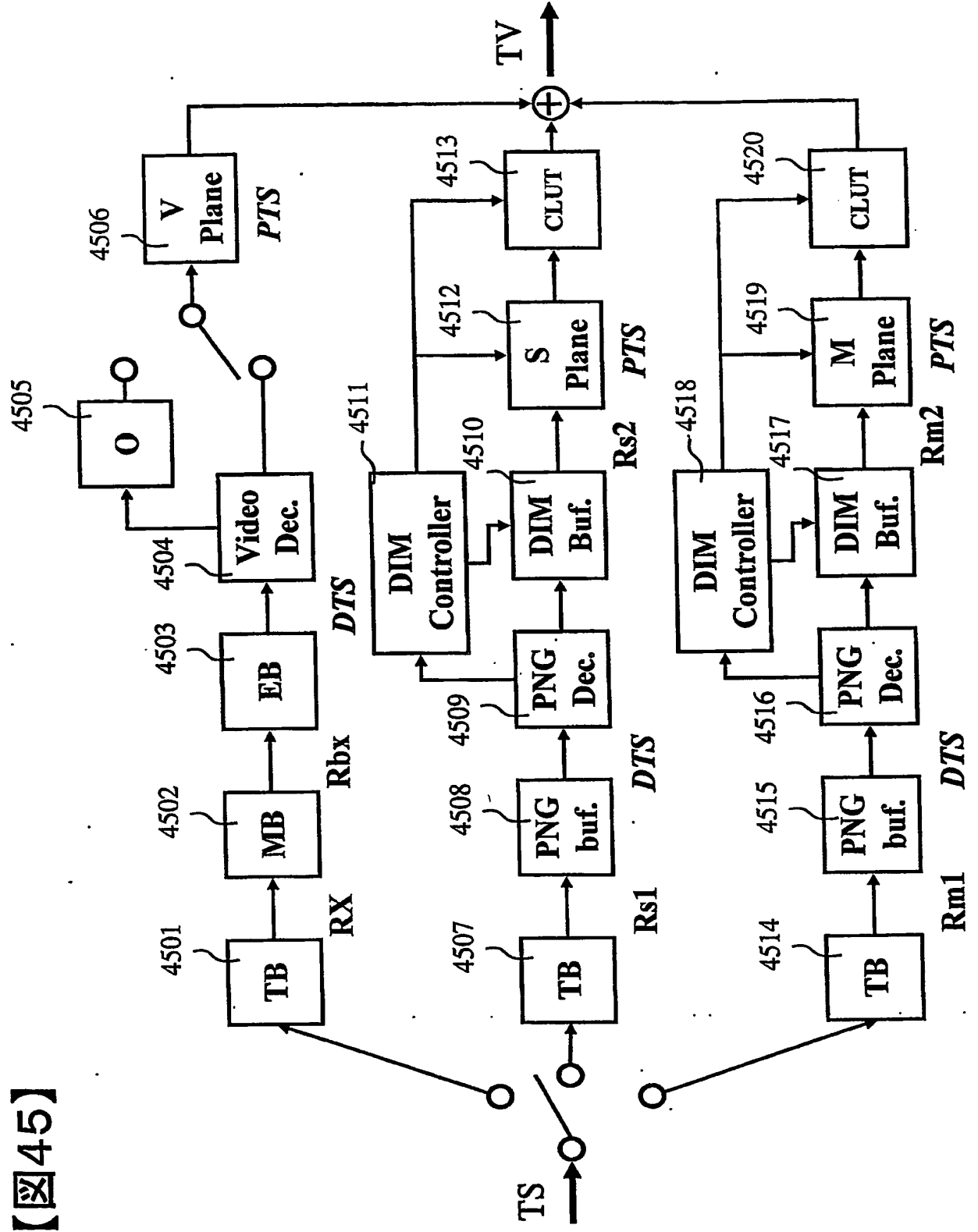
【図44】

BD-Highlight				
PTS0/DTS0	PTS1/DTS1	PTSm/DTSm		
PES packet#0	PES packet#1	...	PES packet#m	
<div> <div>ボタン制御情報</div> <div>共通CLUT</div> <div>PNG#1</div> <div>...</div> <div>PNG#k</div> </div>				

PES packet#0 : 最初に表示されるBDハイライトのイメージデータ群を格納  
PES packet#1 : 1回目のユーザ操作により表示されるイメージデータ群を格納

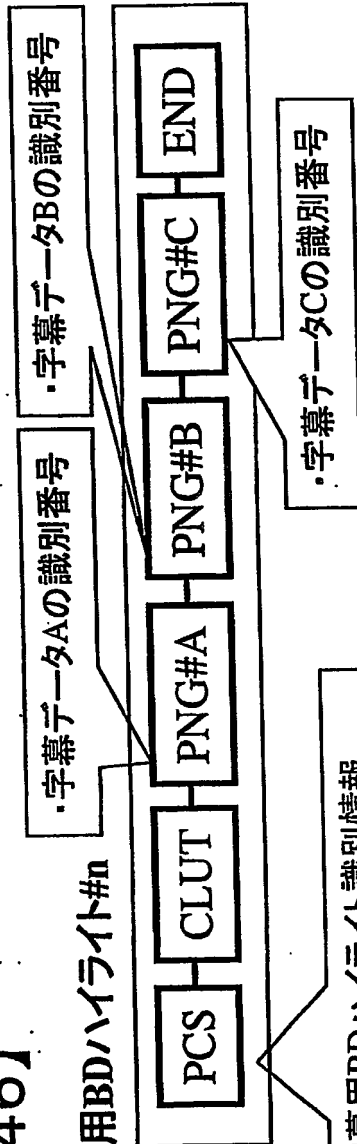
$PTS_{i+1} = PTS_i + \text{frame presentation period}$   
 $DTS_{i+1} = DTS_i + \text{frame presentation period}, (i=0,1,2,\dots,m-1)$

【図45】



【図46】

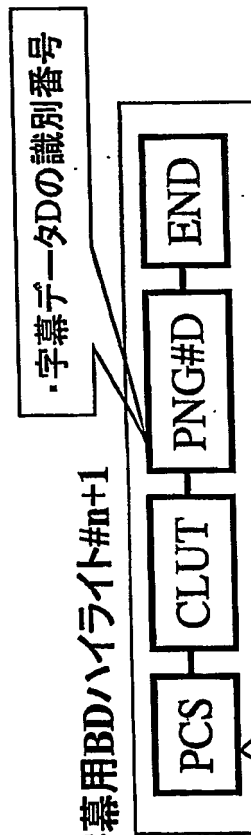
字幕用BDハイライト#n



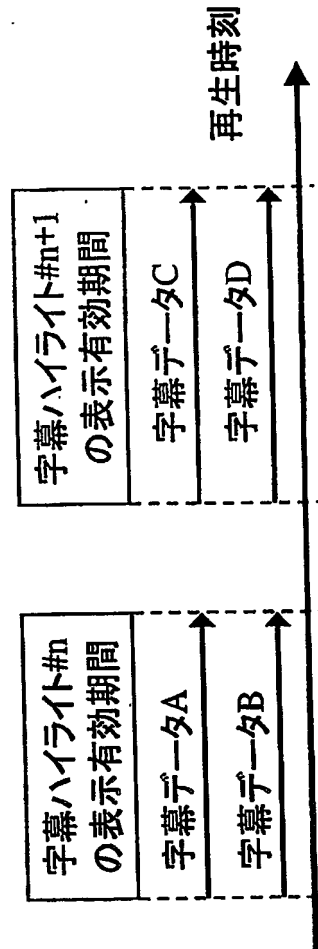
- ・字幕用BDハイライト識別情報
- ・字幕データ(A,B)の識別番号
- ・字幕データ(A,B)の表示位置
- ・字幕データ(A,B)の表示終了時刻

字幕データCは、後続のPCSで制御される

字幕用BDハイライト#n+1

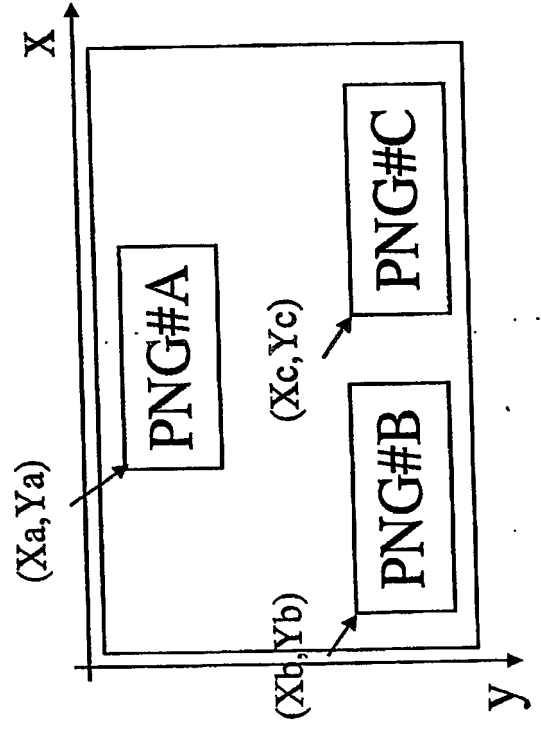
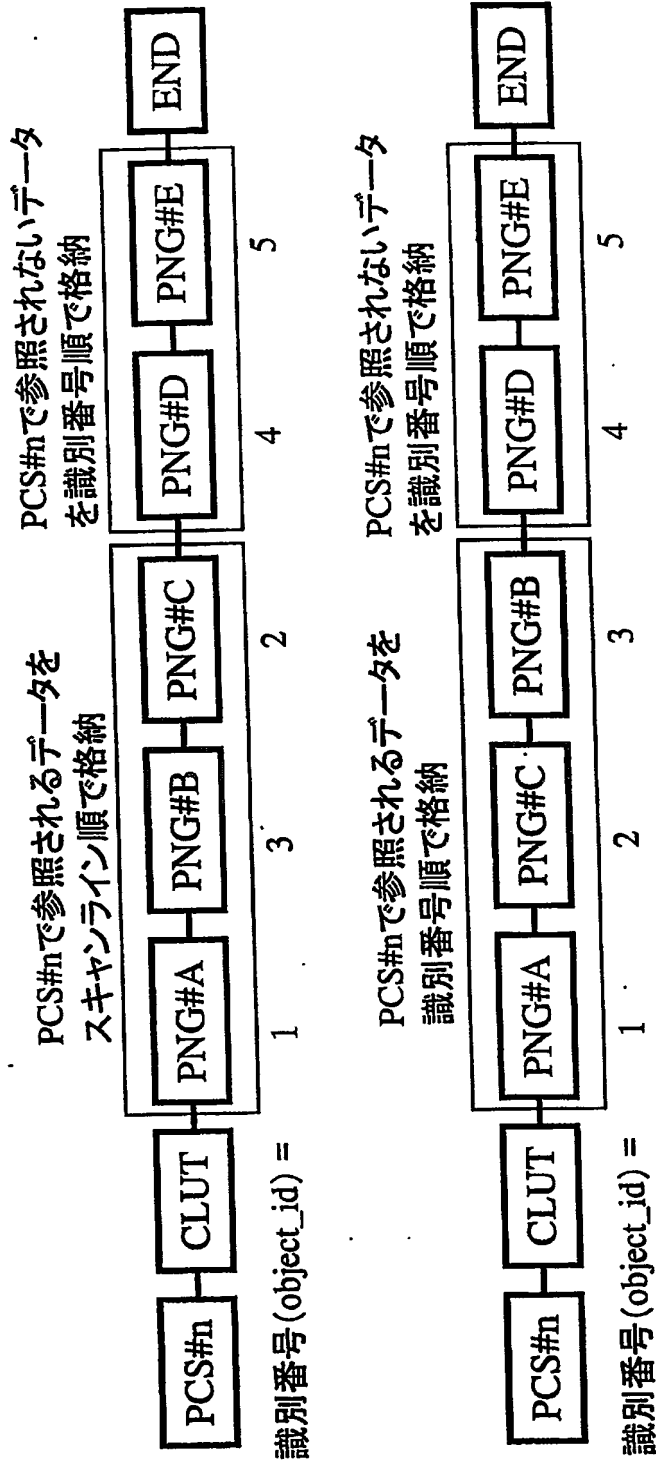


- ・字幕用BDハイライト識別情報
- ・字幕データ(C,D)の識別番号
- ・字幕データ(C,D)の表示位置
- ・字幕データ(C,D)の表示終了時刻

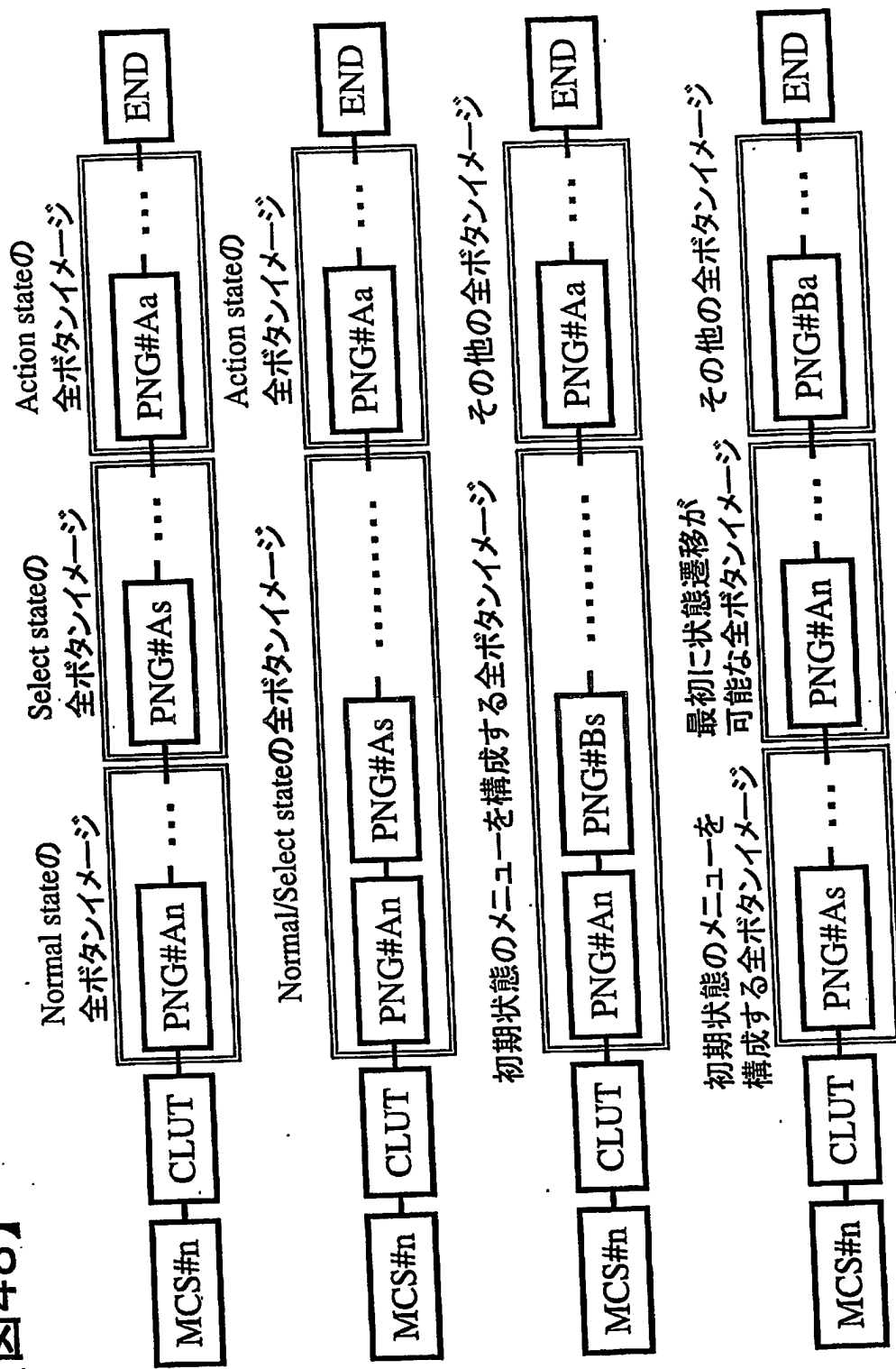




【図47】



【図48】



各グループ(二重枠)内のボタンイメージは、次のどちらかで格納される

- ① スキャンライン順＋スキャンライン同順のイメージは識別番号(object\_id)順
- ② 識別番号(object\_id)順